

**PENGEMBANGAN *CUSTOMER RELATIONSHIP*
MANAGEMENT (CRM) SISTEM INFORMASI
E-COMPLAINT KOTA XYZ**

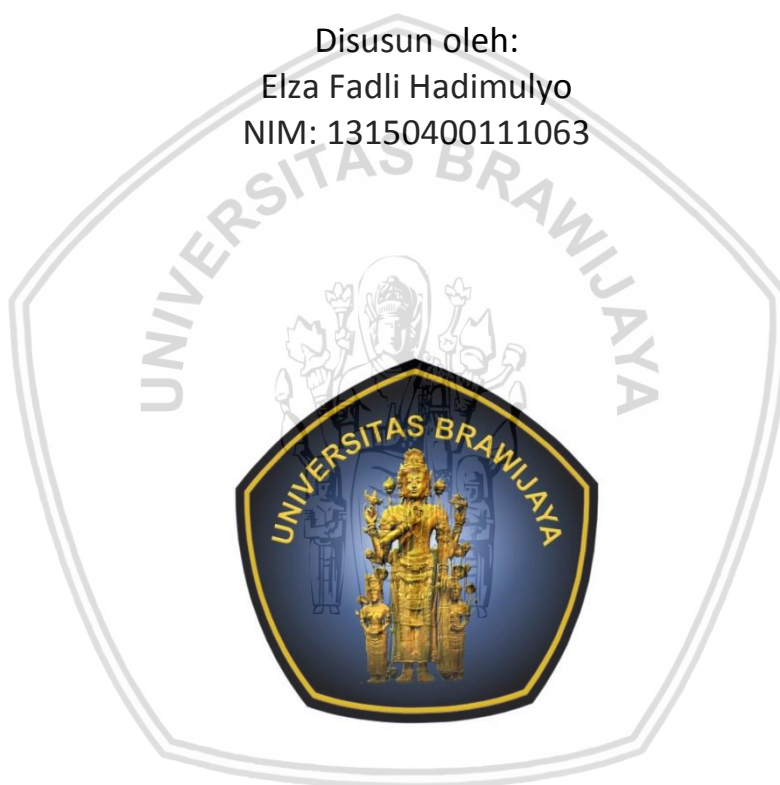
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Elza Fadli Hadimulyo

NIM: 13150400111063



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN *CUSTOMER RELATIONSHIP MANAGEMENT* (CRM)
SISTEM INFORMASI *E-COMPLAINT* KOTA XYZ

SKRIPSI

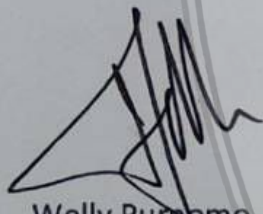
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Elza Fadli Hadimulyo
NIM: 135150400111063

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2018

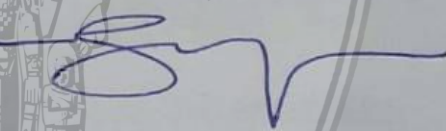
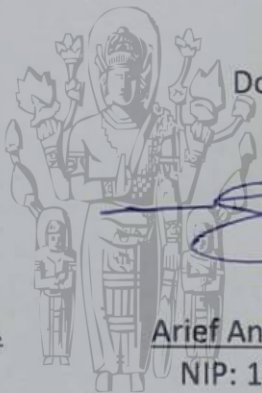
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Welly Purnomo, S.T., M.Kom.
NIP: 201708 810117 1 001

Dosen Pembimbing II



Arief Andy Soebroto, S.T., M.Kom
NIP: 19720425 199903 1 002

Mengetahui

Ketua Jurusan Sistem Informasi



Herman Tolle, Dr. Eng., S.T, M.T
NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 31 Juli 2018



Elza Fadli Hadimulyo

NIM: 135150400111063

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan *Customer Relationship Management* (CRM) Sistem Informasi E-Complaint Kota XYZ”.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Welly Purnomo, S.T., M.Kom. selaku dosen pembimbing I yang telah memberikan ilmu, kritik, serta saran dalam penyusunan skripsi ini.
2. Bapak Arief Andy Soebroto, S.T, M.Kom selaku dosen pembimbing II yang juga telah memberikan ilmu, kritik, serta saran dalam penyusunan skripsi ini.
3. Bapak Sunari dan Ibu Ayu Sa'in selaku orang tua dari penulis, yang selama ini telah sangat berjasa dalam membimbing dan membesarkan penulis, selalu mencurahkan kasih sayang, doa, serta berbagai dukungan moral maupun materi kepada penulis.
4. Seluruh Dosen dan seluruh civitas akademika Program Teknologi Informasi dan Ilmu Komputer yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya.
5. Selvina Yustriasanti yang telah memberikan kesempatan untuk melakukan wawancara demi kelengkapan data-data yang dibutuhkan serta menerima penulis untuk dapat melaksanakan pengerjaan skripsi.
6. Seluruh Keluarga Besar Mahasiswa Sistem Informasi (KBMSI) yang telah memberikan pengalaman serta semangat dan dukungan dalam penulisan skripsi ini.
7. Seluruh teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah memberikan semangat dan dukungan agar skripsi ini cepat terselesaikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 31 Juli 2018

Penulis

elzafadli@gmail.com

ABSTRAK

Belum adanya penggunaan sistem informasi pada pengelolaan dan penanganan pengaduan di kota XYZ mengakibatkan proses pengelolaan dan penanganan pengaduan yang masuk membutuhkan waktu yang lebih lama. Hal ini juga berdampak pada proses pembuatan laporan pengaduan yang menjadi tidak efisien. Untuk meningkatkan kualitas pelayanan pengaduan di kota XYZ, dibutuhkan dukungan sistem informasi untuk mendukung proses pengelolaan dan penanganan pengaduan di kota XYZ dengan menerapkan prinsip-prinsip CRM. Penelitian ini dilakukan untuk mengetahui sistem informasi yang dibangun dapat meningkatkan kualitas pelayanan pengaduan di kota XYZ. Sistem informasi akan dibangun menggunakan metode Waterfall dan pemodelan berorientasi objek. Untuk membangun sistem informasi tersebut dilakukan analisis kebutuhan, perancangan, implementasi, dan pengujian. Tahapan analisis kebutuhan menghasilkan penggambaran proses bisnis menggunakan BPMN dan use case diagram untuk memodelkan kebutuhan dan aktor yang terlibat dalam sistem. Tahapan perancangan menghasilkan activity diagram, dokumentasi model interaksi objek ke dalam bentuk sequence diagram, model objek class diagram, perancangan basis data, perancangan algoritme, dan perancangan antarmuka. Pada tahap implementasi dihasilkan sistem informasi sesuai dengan perancangan yang dibuat pada tahap sebelumnya. Setelah itu dilakukan pengujian white box menggunakan basis path testing untuk mengetahui kompleksitas kode program yang dibuat dan pengujian black box testing untuk menguji kebutuhan fungsional yang dibuat sudah berjalan sesuai dengan yang diharapkan. Untuk menguji sistem informasi yang dibangun dapat meningkatkan kualitas pelayanan pengaduan di kota XYZ dilakukan pengujian UAT. Hasil dari penelitian ini dapat disimpulkan sistem informasi yang dibangun dapat meningkatkan kualitas pelayanan pengaduan di kota XYZ.

Kata kunci: *sistem informasi, waterfall, pemodelan berorientasi objek, customer relationship management (CRM), pengaduan.*

ABSTRACT

The absence of the use of information systems in the management and handling of complaints in the city of XYZ resulted in the management and handling of incoming complaints takes a longer time. This also affects the process of making reports of complaints becoming inefficient. To improve the quality of complaints services in XYZ city, it is necessary to support the information system to support the management and handling complaints in XYZ city by applying the principles of CRM. This research was conducted to find out the information system built can improve the quality of complaint service in XYZ city. The information system will be built using Waterfall method and object oriented modeling. To build information systems researchers needs requirement analysis, design, implementation, and testing. Stages of needs analysis result in the depiction of business processes using BPMN and use case diagrams to model the needs and actors involved in the system. Stages of design produce activity diagrams, documentation of object interaction models into sequence diagrams, object model class diagrams, database design, algorithm design, and interface design. In the implementation stage generated information system in accordance with the design made in the previous stage. After that do white box testing using basis path testing to find out the complexity of the program code created and black box testing to test the functional requirements that are made is running as expected. To test the information system built can improve the quality of complaints service in XYZ city conducted UAT testing. The results of this study can be concluded that the information system built can improve the quality of complaints services in XYZ city.

Keywords: information system, waterfall, object-oriented modeling, customer relationship management (CRM), complaint.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	4
1.3 Tujuan.....	4
1.4 Manfaat.....	4
1.5 Batasan masalah	5
1.6 Sistematika pembahasan	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Pengaduan <i>Online</i>	8
2.3 Customer Relationship Management (CRM)	9
2.4 Manfaat CRM	11
2.5 Pengertian Sistem Informasi	12
2.6 Metode Pengembangan Model <i>Waterfall</i>	12
2.7 Proses Bisnis	13
2.8 <i>Bussines Processs Model Notation</i>	13
2.8.1 <i>Flow Object</i>	13
2.8.2 <i>Connections</i>	15
2.8.3 <i>Swimlanes</i>	16
2.8.4 <i>Artifacts</i>	17
2.9 Analisis Kebutuhan	17

2.9.1	<i>Business Perspective</i>	17
2.9.2	<i>User Perspective</i>	20
2.10	<i>Code Igniter Framework</i>	21
2.11	<i>Unified Model Language (UML)</i>	22
2.12	<i>Use case diagram</i>	23
2.13	<i>Activity diagram</i>	24
2.14	<i>Sequence diagram</i>	25
2.15	<i>Class Diagram</i>	27
2.16	<i>Physical Data Model</i>	29
2.17	Pengujian <i>White-Box</i>	30
2.18	Pengujian <i>Black-Box</i>	31
BAB 3	METODOLOGI	33
3.1	Studi Literatur	33
3.2	Pengumpulan Data	34
3.2.1	Wawancara	34
3.2.2	Observasi	34
3.3	Metode Pengembangan Sistem	34
3.3.1	Analisis Kebutuhan	34
3.3.2	Perancangan Sistem	35
3.3.3	Implementasi	36
3.3.4	Pengujian	37
3.3.5	Pengambilan Kesimpulan dan Saran	37
BAB 4	ANALISIS KEBUTUHAN	38
4.1	<i>Business Perspective</i>	38
4.1.1	Identifikasi Proses Bisnis <i>As-Is</i>	38
4.1.2	Analisis Permasalahan	39
4.1.3	Pemodelan Proses Bisnis <i>To-Be</i>	41
4.1.4	Analisis Kebutuhan Pemangku Kepentingan	46
4.2	<i>User Prespective</i>	50
4.2.1	Identifikasi Aktor	51
4.2.2	Kebutuhan Fungsional dan Non-Fungsional	51
4.3	<i>Use Case Diagram</i>	53

4.4	Spesifikasi <i>Use Case</i>	54
4.4.1	Spesifikasi <i>Use case</i> Mengirim Pengaduan	55
4.4.2	Spesifikasi <i>Use case</i> Verifikasi Pengaduan	55
4.4.3	Spesifikasi <i>Use case</i> Menyalurkan Pengaduan.....	56
4.4.4	Spesifikasi <i>Use case</i> Mengidentifikasi Pengaduan	57
4.4.5	Spesifikasi <i>Use case</i> Mendelegasikan Petugas Lapangan	58
4.4.6	Spesifikasi <i>Use case</i> Mengirim Laporan Penanganan	58
4.4.7	Spesifikasi <i>Use case</i> Mengkonfirmasi Laporan Penanganan....	59
4.4.8	Spesifikasi <i>Use case</i> Menanggapi Pengaduan	60
4.4.9	Spesifikasi <i>Use case</i> Melihat Pesan	61
4.4.10	Spesifikasi <i>Use case</i> Melihat Status Pengaduan.....	61
4.4.11	Spesifikasi <i>Use case</i> Mencari Laporan Pengaduan.....	62
4.4.12	Spesifikasi <i>Use case</i> Melihat <i>Dashboard</i> Pengaduan	62
4.4.13	Spesifikasi <i>Use case</i> Membuat Laporan	63
4.4.14	Spesifikasi <i>Use case</i> Melihat Laporan Pengaduan	63
4.4.15	Spesifikasi <i>Use case</i> Registrasi.....	64
4.4.16	Spesifikasi <i>Use case</i> Mengelola Akun.....	64
4.4.17	Spesifikasi <i>Use case</i> Login	65
4.4.18	Spesifikasi <i>Use case</i> Logout	66
4.4.19	Spesifikasi <i>Use case</i> Memperbarui Akun	66
BAB 5	PERANCANGAN.....	68
5.1	<i>Activity diagram</i>	68
5.1.1	<i>Activity diagram</i> Mengirim Pengaduan	68
5.1.2	<i>Activity diagram</i> Verifikasi Pengaduan 3b: Tidak Memenuhi Kriteria	69
5.1.3	<i>Activity diagram</i> Menyalurkan Pengaduan.....	70
5.1.4	<i>Activity diagram</i> Menanggapi Pengaduan	71
5.1.5	<i>Activity diagram</i> Mendelegasikan Petugas Lapangan.....	72
5.2	<i>Sequence diagram</i>	73
5.2.1	<i>Sequence diagram</i> Mengirim Pengaduan	73
5.2.2	<i>Sequence diagram</i> Menyalurkan Pengaduan.....	74
5.2.3	<i>Sequence diagram</i> Menanggapi Pengaduan.....	74

5.2.4	<i>Sequence diagram</i> Mendelegasikan Petugas Lapangan	75
5.3	Perancangan <i>Class Diagram</i>	76
5.4	Perancangan Basis Data	78
5.5	Perancangan Antarmuka.....	80
5.5.1	Halaman Awal.....	80
5.5.2	Halaman Beranda <i>User</i>	81
5.5.3	Halaman Pengaduan <i>Admin</i>	81
5.5.4	Halaman Pengaduan <i>User</i>	82
5.6	Perancangan Algoritme.....	83
5.6.1	Algoritme Mengirim Pengaduan	83
5.6.2	Algoritme Verifikasi Pengaduan	84
BAB 6	IMPLEMENTASI	85
6.1	Spesifikasi Sistem	85
6.2	Implementasi <i>Class Diagram</i>	85
6.2.1	Implementasi <i>class diagram</i> kelas pengaduan	85
6.2.2	Implementasi <i>class diagram</i> kelas db_pengaduan	86
6.3	Implementasi Algoritme.....	88
6.3.1	Verifikasi Pengaduan.....	88
6.3.2	Mengirim Pengaduan ke OPD	89
6.4	Implementasi Basis Data.....	90
6.4.1	Implementasi Tabel Pengaduan.....	90
6.4.2	Implementasi Tabel Users	91
6.5	Implementasi Antarmuka	91
6.5.1	Halaman Awal.....	92
6.5.2	Halaman Mengirim Pengaduan.....	92
6.5.3	Halaman Melihat Status Pengaduan	93
6.5.4	Halaman Verifikasi Pengaduan.....	93
6.5.5	Halaman Membalas Pengaduan	94
6.5.6	Halaman Membuat Laporan	95
BAB 7	PENGUJIAN	96
7.1	Pengujian <i>White-Box</i>	96
7.1.1	Pengujian Basis Path Verifikasi Pengaduan.....	96

7.2	Pengujian <i>Black-Box</i>	98
7.3	User Acceptance Testing (UAT)	101
BAB 8	PENUTUP	105
8.1	Kesimpulan	105
8.2	Saran	106
DAFTAR PUSTAKA	107
LAMPIRAN A HASIL WAWANCARA	108
LAMPIRAN B HASIL PENGUJIAN UAT	113



DAFTAR TABEL

Tabel 2.1 Tipe <i>event</i>	14
Tabel 2.2 Tipe <i>activity</i>	14
Tabel 2.3 Tipe <i>gateway</i>	15
Tabel 2.4 Tipe <i>connections</i>	16
Tabel 2.5 Tipe <i>swimlanes</i>	16
Tabel 2.6 Tipe <i>artifacts</i>	17
Tabel 2.7 Analisis permasalahan	18
Tabel 2.8 Analisis tipe pemangku kepentingan	19
Tabel 2.9 Peran dan perwakilan pemangku kepentingan	20
Tabel 2.10 Simbol-simbol <i>use case diagram</i>	23
Tabel 2.11 Simbol-simbol <i>activity diagram</i>	24
Tabel 2.12 Simbol-simbol <i>sequence diagram</i>	26
Tabel 2.13 Simbol-simbol diagram <i>class</i>	28
Tabel 4.1 Analisis permasalahan	39
Tabel 4.2 Pemangku Kepentingan dan Tipe Pemangku Kepentingan	46
Tabel 4.3 Peran dari Perwakilan Pemangku Kepentingan	47
Tabel 4.4 Kebutuhan pemangku kepentingan	48
Tabel 4.5 Identifikasi aktor	51
Tabel 4.6 Kebutuhan Fungsional	52
Tabel 4.7 Spesifikasi <i>use case</i> mengirim pengaduan	55
Tabel 4.8 Spesifikasi <i>use case</i> verifikasi pengaduan	55
Tabel 4.9 Spesifikasi <i>use case</i> menyalurkan pengaduan	56
Tabel 4.10 Spesifikasi <i>use case</i> mengidentifikasi pengaduan	57
Tabel 4.11 Spesifikasi <i>use case</i> mendelegasikan petugas lapangan	58
Tabel 4.12 Spesifikasi <i>use case</i> mengirim laporan penanganan	58
Tabel 4.13 Spesifikasi <i>use case</i> mengkonfirmasi laporan penanganan	59
Tabel 4.14 Spesifikasi <i>use case</i> menanggapi pengaduan	60
Tabel 4.15 Spesifikasi <i>use case</i> melihat pesan	61
Tabel 4.16 Spesifikasi <i>use case</i> melihat status pengaduan	61
Tabel 4.17 Spesifikasi <i>use case</i> mencari laporan pengaduan	62

Tabel 4.18 Spesifikasi <i>use case</i> melihat <i>dashboard</i> pengaduan	62
Tabel 4.19 Spesifikasi <i>use case</i> membuat laporan.....	63
Tabel 4.20 Spesifikasi <i>use case</i> melihat laporan pengaduan	63
Tabel 4.21 Spesifikasi <i>use case</i> registrasi	64
Tabel 4.22 Spesifikasi <i>use case</i> mengelola akun	64
Tabel 4.23 Spesifikasi <i>use case login</i>	65
Tabel 4.24 Spesifikasi <i>use case logout</i>	66
Tabel 4.25 Spesifikasi <i>use case</i> memperbarui akun.....	66
Tabel 5.1 Perancangan algoritme mengirim pengaduan	84
Tabel 5.2 Perancangan algoritme verifikasi pengaduan	84
Tabel 6.1 Tabel spesifikasi minimum <i>web server</i>	85
Tabel 6.2 Implementasi <i>class diagram</i> tabel pengaduan	85
Tabel 6.3 Implementasi <i>class diagram</i> tabel db_pengaduan	86
Tabel 6.4 Implementasi verifikasi pengaduan	88
Tabel 6.5 Implementasi mengirim pengaduan ke OPD	89
Tabel 6.6 <i>Source code</i> tabel pengaduan	90
Tabel 6.7 <i>Source code</i> tabel users.....	91
Tabel 7.1 Perancangan algoritme verifikasi pengaduan	96
Tabel 7.2 <i>Test case</i> verifikasi pengaduan.....	98
Tabel 7.3 Pengujian Validasi.....	98
Tabel 7.4 Tabel Pertanyaan UAT	101
Tabel 7.5 Pengujian UAT admin 1	102
Tabel 7.6 Pengujian UAT admin 2	103

DAFTAR GAMBAR

Gambar 2.1 Alur pengaduan masyarakat	9
Gambar 2.2 <i>Waterfall model</i>	12
Gambar 2.3 Arsitektur <i>Code Igniter Framework</i>	22
Gambar 2.4 <i>Physical data model</i>	30
Gambar 3.1 Alur metode penelitian	33
Gambar 4.1 Diagram BPMN proses bisnis pengelolaan dan penanganan	39
Gambar 4.2 Diagram BPMN proses bisnis pengelolaan dan penanganan pengaduan sistem baru.....	41
Gambar 4.3 <i>Use case diagram</i>	54
Gambar 5.1 <i>Activity diagram</i> mengirim pengaduan.....	68
Gambar 5.2 <i>Activity diagram</i> verifikasi pengaduan 3b; tidak memenuhi kriteria	69
Gambar 5.3 <i>Activity diagram</i> menyalurkan pengaduan.....	70
Gambar 5.4 <i>Activity diagram</i> menanggapi pengaduan	71
Gambar 5.5 <i>Activity diagram</i> mendelegasikan petugas lapangan	72
Gambar 5.6 <i>Sequence diagram</i> mengirim pengaduan	73
Gambar 5.7 <i>Sequence diagram</i> menyalurkan pengaduan.....	74
Gambar 5.8 <i>Sequence diagram</i> menanggapi pengaduan.....	75
Gambar 5.9 <i>Sequence diagram</i> mendelegasikan petugas lapangan	76
Gambar 5.10 <i>Class diagram</i> sebagai domain controller.....	77
Gambar 5.11 <i>Class diagram</i> sebagai domain model	78
Gambar 5.12 <i>Entity Relational Diagram (ERD)</i>	78
Gambar 5.13 <i>Physical Data Model</i>	79
Gambar 5.14 Perancangan halaman awal	80
Gambar 5.15 Perancangan halaman beranda <i>user</i>	81
Gambar 5.16 Perancangan halaman pengaduan <i>admin</i>	82
Gambar 5.17 Perancangan halaman pengaduan <i>user</i>	83
Gambar 6.1 Implementasi antarmuka halaman awal	92
Gambar 6.2 Implementasi antarmuka halaman mengirim pengaduan	92
Gambar 6.3 Implementasi antarmuka halaman melihat status pengaduan	93
Gambar 6.4 Implementasi antarmuka halaman verifikasi pengaduan.....	94

Gambar 6.5 Implementasi antarmuka halaman membalas pengaduan	94
Gambar 6.6 Halaman membuat laporan	95
Gambar 6.7 Hasil sortir dalam format excel.....	95
Gambar 7.1 <i>Flow graph</i> verifikasi pengaduan	97



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	108
LAMPIRAN B HASIL PENGUJIAN UAT	113



BAB 1 PENDAHULUAN

1.1 Latar belakang

Penggunaan sistem informasi untuk membantu kinerja organisasi semakin dibutuhkan. Dengan dukungan oleh kecanggihan teknologi informasi, telah memungkinkan pengembangan sistem informasi yang semakin handal. Informasi merupakan salah satu sumber daya paling penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Penataan informasi yang dilakukan secara teratur, jelas tepat dan cepat serta dapat disajikan dalam sebuah laporan tentunya sangat mendukung kelancaran kegiatan operasional organisasi dan pengambilan keputusan yang tepat.

Pemerintah Daerah merupakan sebuah organisasi yang bergerak di bidang pemerintahan. Pemerintahan Daerah adalah penyelenggaraan urusan pemerintahan oleh Pemerintah Daerah dan Dewan Perwakilan Rakyat Daerah menurut asas otonomi dan tugas pembantuan dengan prinsip otonomi seluas-luasnya dalam sistem dan prinsip Negara Kesatuan Republik Indonesia sebagaimana dimaksud dalam Undang-Undang Dasar Negara Republik Indonesia Tahun 1945. Pemerintahan Daerah di Indonesia terdiri dari Pemerintahan Daerah Provinsi dan Pemerintahan Daerah Kabupaten/Kota yang terdiri atas kepala daerah dan Dewan Perwakilan Rakyat Daerah (DPRD) dibantu oleh Perangkat Daerah. Pemerintah daerah mempunyai beberapa fungsi yaitu fungsi Regulasi, fungsi Pelayanan, fungsi Pemberdayaan, dan fungsi Pembangunan. Untuk menjalankan fungsi-fungsi tersebut diperlukan peran aktif dari partisipasi masyarakat sehingga pemerintah dapat memetakan berbagai kebutuhan dan masalah yang ada di masyarakat. Oleh karena itu partisipasi masyarakat dalam pelayanan publik perlu ditingkatkan. Pengaduan masyarakat bisa dipandang sebagai bentuk partisipasi dan rasa kepedulian masyarakat dalam pelaksanaan pelayanan publik.

Pengaduan merupakan proses penyampaian informasi yang berisi keluhan atau ketidakpuasan. Pengadu atau pelapor adalah perseorangan atau kelompok yang menyampaikan keluhan atau ketidakpuasan terkait dengan pelaksanaan fungsi pemerintahan. Aduan masyarakat adalah bentuk dari pengawasan yang dilakukan oleh masyarakat yang disampaikan kepada yang diberikan kewenangan untuk menerima dan menindaklanjuti aduan masyarakat. Bentuk pengaduan masyarakat terbagi menjadi dua yaitu pengaduan secara langsung dan pengaduan secara tidak langsung. Pengaduan secara langsung dilakukan oleh masyarakat dengan cara datang langsung ke meja pengaduan atau bertemu langsung dengan pejabat yang berwenang dalam menangani pengaduan masyarakat. Sedangkan pengaduan secara tidak langsung dilakukan oleh masyarakat dengan cara tidak langsung berhadapan atau bertemu dengan pejabat yang berwenang dalam menangani pengaduan masyarakat atau datang langsung ke meja pengaduan/saluran.

Menurut survei yang dilakukan oleh Bappenas tahun 2010 terkait pengaduan atas pelayanan publik menunjukkan bahwa mayoritas masyarakat (52%) menganggap pengaduan tidak berguna bagi mereka. Disisi lain 16% masyarakat menganggap bahwa pemerintah tidak menyediakan sarana pengaduan yang memadai. Sementara 29,1% sisanya beranggapan pengaduan yang mereka sampaikan tidak akan ditindaklanjuti oleh pemberi pelayanan. Berdasarkan hasil survei tersebut mengindikasikan bahwa keengganan masyarakat untuk menyampaikan keluhan atau pengaduan disebabkan oleh adanya keraguan dan ketidakpercayaanya masyarakat akan tindak lanjut dari pengaduan tersebut. Sebagai contoh, di daerah-daerah di Indonesia banyak dijumpai pemerintah menyediakan sarana keluhan dan aduan. Akan tetapi sarana tersebut tidak jelas pengelolaan dan mekanismenya. Penanganan pengaduan yang baik adalah yang dapat memudahkan pemerintah juga masyarakat dalam mengakses pengaduan tersebut. Pengaduan harus dikelola dengan baik agar bisa menjangkau masyarakat, sehingga masyarakat akan responsif terhadap berbagai kebijakan yang dikeluarkan pemerintah.

Pemerintah Kota XYZ menyediakan beberapa sarana untuk menampung pengaduan dari masyarakat seperti penyediaan kotak saran di setiap OPD dan penggunaan media sosial seperti *twitter* atau *facebook*. Pemerintah juga menyediakan layanan pengaduan untuk masyarakat jika ingin melaporkan pengaduan secara langsung. Untuk pelaporan melalui layanan pengaduan, masyarakat harus mengisi formulir identitas terlebih dahulu kemudian masyarakat menyampaikan keluhan atau ketidakpuasan mereka. Laporan pengaduan yang masuk akan direkap oleh petugas pelayanan pengaduan pada suatu buku dan kemudian dilakukan identifikasi apakah pengaduan perlu dilakukan penanganan lebih lanjut atau tidak. Jika laporan pengaduan perlu dikirim ke OPD lain untuk ditindaklanjuti maka pihak pelayanan pengaduan akan menghubungi OPD tersebut dan menyampaikan permasalahan pengaduan.

Pada proses pelaporan pengaduan saat ini terdapat beberapa permasalahan seperti pada pelaporan pengaduan melalui kotak saran terdapat permasalahan untuk mengidentifikasi beberapa pengaduan disebabkan laporan pengaduan tidak jelas seperti identitas pengirim maupun masalah pengaduan yang disampaikan. Begitu juga dengan pelaporan melalui media sosial, pihak pengelola pengaduan kesulitan mengidentifikasi identitas pelapor. Sedangkan pada proses pengelolaan pengaduan, pihak pengelola pengaduan kesulitan untuk mengetahui laporan yang dikirim ke OPD lain sudah ditindaklanjuti atau tidak. Masyarakat juga kesulitan untuk mengetahui laporan mereka sudah ditindaklanjuti atau tidak dikarenakan masyarakat harus menghubungi pihak pelayanan pengaduan untuk mengetahui hasil dari pengaduan yang mereka sampaikan. Masalah lain muncul ketika pihak pengelola pengaduan ingin melakukan klasifikasi dan kategorisasi pengaduan karena pihak pengaduan harus menginputkan lagi data pengaduan yang masuk ke *Microsoft Excel* untuk bisa diklasifikasi dan dikategorisasi. Permasalahan berikutnya pihak pengelola pengaduan membutuhkan waktu yang lama untuk membuat laporan pengaduan dikarenakan pihak pengelola pengaduan harus mengolah lagi data yang ada di *Microsoft Excel* untuk diproses

menjadi sebuah laporan. Tidak adanya sistem yang mengintegrasikan data pengaduan yang masuk pada masing-masing OPD juga menimbulkan masalah untuk mengelola data pengaduan. Oleh sebab itu, maksud dari penelitian ini adalah mengembangkan sistem informasi berbasis *web* untuk melakukan pengolahan dan penanganan pengaduan di kota XYZ. Sistem informasi ini dibangun berbasis *web* sehingga pengguna aplikasi diharapkan lebih mudah mengakses informasi yang diberikan dan bisa diakses secara *online* dengan menggunakan bermacam-macam perangkat. Dari permasalahan tersebut, maka Pemerintah Kota XYZ berencana membuat sistem informasi *E-Complaint* berbasis *web*.

Sistem informasi *E-Complaint* termasuk dalam kategori *Operational Customer Relationship Management* (CRM) berjenis *Service Automation* karena menurut Buttle (Buttle 2008) *Operational CRM* adalah mengotomatisasi dan mengembangkan proses-proses bisnis *customer-facing* dan *customer-supporting*. Software aplikasi CRM memungkinkan menjalankan servis yang telah terotomatisasi dan terintegrasi. Sistem informasi *E-Complaint* membantu masyarakat turut aktif dalam proses pelayanan pengaduan dengan tujuan untuk memudahkan pengelolaan pengaduan dan mempercepat penanganan pengaduan.

Metode pengembangan yang akan digunakan untuk membangun sistem ini menggunakan metode *Waterfall* dan model pengembangan menggunakan pendekatan *object oriented*. Menurut Sommerville (2011) metode pengembangan *Waterfall* adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Dalam proses implementasi metode *Waterfall* ini, sebuah tahapan akan diselesaikan terlebih dahulu dimulai dari tahapan yang pertama sebelum melanjutkan ke tahapan yang berikutnya. Keuntungan dari penggunaan metode *Waterfall* ini yaitu kebutuhan harus didefinisikan lebih mendalam sebelum proses pembuatan kode program dilakukan. Metode *Waterfall* ini juga memungkinkan sedikit mungkin perubahan yang dilakukan saat pembuatan sistem berlangsung.

Menurut Sommerville (2011) sistem yang dikembangkan menggunakan pendekatan *object oriented* lebih mudah menghadapi perubahan daripada sistem yang dikembangkan dengan pendekatan fungsional. Ini dikarenakan pada pendekatan *object oriented* mengubah atau menambahkan layanan di suatu objek seharusnya tidak akan berdampak pada objek sistem lainnya. Dalam pengembangan sistem menggunakan pendekatan *object oriented* terdapat proses pemodelan sistem. Proses pemodelan sistem merupakan suatu proses untuk mengembangkan model abstraks dari suatu sistem dengan setiap model menyajikan pandangan atau prespektif berbeda dari sistem. Dalam proses pemodelan sistem beberapa notasi grafis dapat digunakan sebagai alat bantu merepresentasikan sistem. Pada pendekatan *object oriented* penggunaan notasi grafis dapat didasarkan pada notasi dalam unified modeling language (UML). Model UML digunakan pada proses analisis kebutuhan untuk membantu

memodelkan kebutuhan sistem dan pada proses perancangan untuk menggambarkan sistem sebagai acuan untuk proses implementasi.

Untuk implementasi sistem dengan menggunakan pendekatan *object oriented* dibutuhkan kerangka kerja aplikasi yang menerapkan pendekatan *object oriented*. Code Igniter merupakan salah satu kerangka kerja aplikasi berbasis PHP yang menerapkan pendekatan *object oriented*. Pada saat proses implementasi, objek yang teridentifikasi pada desain diimplementasikan sebagai komponen sistem. Komponen tersebut bertanggung jawab untuk menyediakan fungsi yang merupakan kemampuan dari sistem untuk memenuhi kebutuhan pengguna. Untuk memastikan bahwa sistem yang telah dibangun memenuhi kebutuhan dan dapat membantu meningkatkan kualitas pelayanan pengaduan di kota XYZ, maka diperlukan pengujian.

Berdasarkan penjelasan tersebut, penelitian ini dilakukan untuk mengembangkan sebuah sistem informasi yang dapat membantu proses pengelolaan dan penanganan pengaduan di kota XYZ dengan melalui tahapan analisis kebutuhan, perancangan, dan implementasi menggunakan pendekatan *object oriented*. Harapan dari penelitian ini adalah agar skripsi ini dapat membantu meningkatkan kualitas pelayanan pengaduan di kota XYZ.

1.2 Rumusan masalah

Berdasarkan latar belakang tersebut, terdapat beberapa permasalahan yang akan dihadapi dalam penelitian dan pengembangan sistem yaitu:

1. Bagaimana proses analisis dan perancangan *E-Complaint* berbasis *web* ?
2. Bagaimana hasil dari implementasi *E-Complaint* sebagai pengolah dan penanganan pengaduan di kota XYZ ?
3. Bagaimana membuat laporan pengarsipan pengaduan sesuai dengan format yang dibutuhkan Pemerintah Kota XYZ ?
4. Apakah terdapat peningkatan kualitas pelayanan dan pengelolaan pengaduan di kota XYZ dengan adanya sistem informasi *E-Complaint*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Menganalisis proses pengelolaan pengaduan yang ada di kota XYZ.
2. Merancang sistem informasi pengaduan yang dapat menunjang aktivitas pengolahan dan penanganan pengaduan di kota XYZ.
3. Mengembangkan sistem informasi pengaduan berbasis *web* di kota XYZ.
4. Meningkatkan kualitas pelayanan dan pengelolaan pengaduan di Kota XYZ.

1.4 Manfaat

Manfaat yang didapat oleh Pemerintah Kota di kota XYZ dari penelitian ini antara lain:

1. Menyediakan sarana pengaduan yang dapat diakses secara online dan mudah.
2. Menyediakan sistem informasi pengelolaan dan penanganan pengaduan yang terintegrasi dan dapat diakses secara *real-time*.
3. Membantu proses pembuatan laporan terkait data pengaduan.

1.5 Batasan masalah

Dalam pengembangan Sistem Informasi Pengaduan ini terdapat beberapa batasan masalah yaitu:

1. Ruang lingkup pelayanan penanganan pengaduan penelitian ini meliputi pengaduan terkait hambatan dalam pelayanan masyarakat, pelanggaran disiplin pegawai, dan pelanggaran terhadap standar pelayanan yang telah ditetapkan.
2. Perancangan penyajian format laporan pengarsipan terkait pengaduan sesuai dengan kebutuhan Pemerintah Kota.
3. Dalam melakukan perancangan sistem informasi digunakan notasi *UML Diagram* yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* sedangkan untuk pemodelan database menggunakan ERD dan PDM.

1.6 Sistematika pembahasan

Dalam penyusunan proposal penelitian ini terdiri dari delapan bab dengan beberapa pokok bahasan. Berikut sistematika penulisan proposal penelitian ini:

BAB 1 PENDAHULUAN

Pada bab ini berisikan uraian latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi uraian tentang penelitian sebelumnya, teori-teori dasar dan teori-teori penunjang yang digunakan dalam penelitian ini.

BAB 3 METODE PENELITIAN

Bab ini berisi metode penelitian yang dilakukan yakni studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, dan terakhir pembuatan kesimpulan.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini terdapat analisis kebutuhan sistem yang dibangun. Pada tahap ini dilakukan pemodelan proses bisnis, penentuan pemangku kepentingan dan user, penentuan kebutuhan fungsional dan non-fungsional, pemodelan *use case* dan spesifikasi *use case*.

BAB 5 PERANCANGAN

Pada bab ini berisi perancangan sistem yang terdiri dari pembuatan *activity diagram*, *sequence diagram*, *class diagram*, perancangan antarmuka, perancangan basis data, dan perancangan algoritme.

BAB 6 IMPLEMENTASI

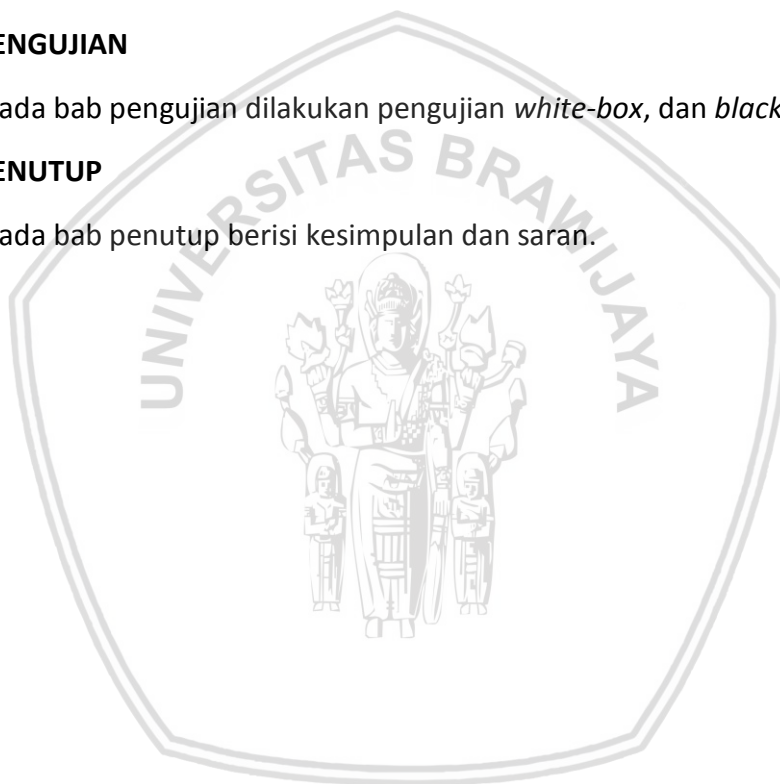
Pada bab ini dijelaskan mengenai spesifikasi implementasi, implementasi *class diagram*, implementasi basis data, dan implementasi antarmuka.

BAB 7 PENGUJIAN

Pada bab pengujian dilakukan pengujian *white-box*, dan *black-box*.

BAB 8 PENUTUP

Pada bab penutup berisi kesimpulan dan saran.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian pertama yang dijadikan sebagai kajian pustaka adalah penelitian yang dilakukan oleh Adzanil Rachmadi Rachmadi P et al. (2017) dengan judul *"Pembangunan Sistem Informasi Manajemen Keluhan Pelanggan Berbasis Webgis"*. Penelitian tersebut membahas tentang membuat sistem informasi untuk menampung keluhan pelanggan dengan tujuan agar pihak outlet McDonald's cabang Watu Gong Malang dapat mengetahui informasi mengenai jumlah komplain dan waktu komplain lebih dini. Dengan adanya sistem informasi tersebut pihak outlet McDonald's cabang Watu Gong Malang dapat mengevaluasi dan melakukan perbaikan terhadap kualitas pelayanan agar lebih adaptif serta lebih cepat berdasarkan informasi yang didapat dari Sistem Informasi Manajemen Keluhan Pelanggan Berbasis WebGIS.

Penelitian kedua dalam kajian pustaka ini adalah penelitian yang dilakukan oleh Mirza Zanuar (Zanuar, Saputra, dan Pradana 2018) dengan judul *"Pengembangan Sistem Informasi Kim (Kelompok Informasi Masyarakat) Kominfo Jatim Berbasis Web"*. Penelitian tersebut membahas tentang bagaimana membuat sistem informasi yang dapat menampung informasi-informasi dari berbagai daerah khususnya dari Jawa Timur. Dalam penelitian ini metode pengembangan yang digunakan metode *Waterfall* dan metode pemodelan yang digunakan adalah *Object Oriented*.

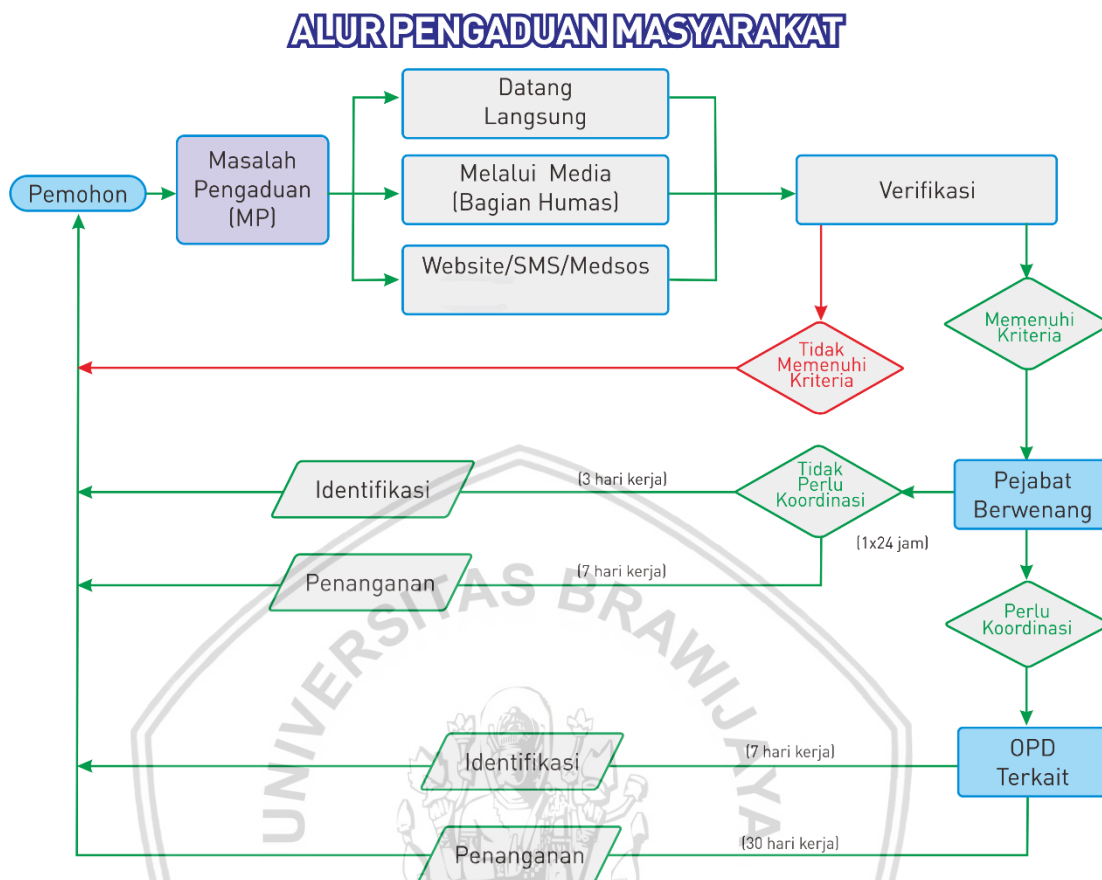
Penelitian ketiga dalam kajian pustaka ini adalah penelitian yang dilakukan oleh (Anofrizen, 2017) dengan judul *"Sistem Informasi Pengaduan Masyarakat Program Keluarga Harapan Kota Pekanbaru"*. Penelitian tersebut membahas tentang Dinas Sosial dan Pemakaman Kota Pekanbaru yang mempunyai sebuah program bantuan untuk meningkatkan pengeluaran rumah tangga sangat miskin yakni Program Keluarga Harapan (PKH). Tujuan dari penelitian ini untuk merancang dan membangun Sistem Informasi Pengaduan Masyarakat Program Keluarga Harapan yang menggunakan *database* sebagai media penyimpanan data yang dapat memudahkan dalam pencarian data, pembuatan dan penyimpanan, dan menggantikan penggunaan sistem pengaduan manual. Metode pengembangan sistem yang digunakan adalah metode pengembangan sistem *Agile* dan untuk perancangan dalam penelitian ini menggunakan *UML*, perancangan *interface* hingga pengkodean menggunakan PHP, hingga sampai pada tahap pengujian fungsional sistem menggunakan *blackbox testing*. Sistem PKH yang dibangun dapat mengatasi permasalahan yang terjadi pada sistem lama terkait pengelolaan data pengaduan Program Keluarga Harapan Kota Pekanbaru.

2.2 Pengaduan *Online*

Pengaduan *online* adalah sarana interaktif masyarakat dan pemerintah berbasis web untuk pengawasan dan pengendalian pembangunan. Berdasarkan Peraturan Walikota XYZ No. XX Tahun YY, kota XYZ memiliki Pedoman Penyelenggaraan Pelayanan Penanganan Pengaduan sebagai berikut :

1. Lingkup Pengaduan
 - Penyalahgunaan wewenang
 - Korupsi – kolusi Nepotisme
 - Pelanggaran standar pelayanan
 - Pelanggaran disiplin
 - Hambatan pelayanan masyarakat
2. Hak Penerima Pelayanan
 - Menyampaikan berupa keluhan, kritik, pernyataan tidak puas
 - Memperoleh tanggapan
 - Mengetahui hasil pengaduan
3. Cara Penyampaian Pengaduan
 - Datang langsung
 - Melalui surat
 - Melalui media cetak/elektronik
 - Website Pemerintah Kota XYZ
 - Layanan SMS Gateway
4. Tugas Pelayanan
 - Menerima pengaduan
 - Registrasi
 - Menindaklanjuti ke instansi pelaksana yang berwenang
 - Menyampaikan hasil penyelesaian ke pemohon
 - menyusun/mengirim laporan ke Sekretariat pengaduan
5. Waktu Penyelesaian Tanpa Perlu Koordinasi
 - Paling lama 3 hari kerja sejak pengaduan diterima harus identifikasi/klarifikasi
 - Paling lama 7 hari kerja sejak pengaduan diterima harus menyampaikan hasil
6. Waktu Penyelesaian Perlu Koordinasi
 - Paling lama 7 hari kerja sejak pengaduan diterima harus identifikasi/klarifikasi
 - Paling lama 30 hari kerja sejak pengaduanditerima harus menyampaikan hasil

Alur Pelayanan Pengaduan :



Gambar 2.1 Alur pengaduan masyarakat

Sumber: Diadaptasi dari DTSS Kepatuhan Internal (2014)

2.3 Customer Relationship Management (CRM)

CRM adalah strategi bisnis utama yang mengintegrasikan proses internal dan jaringan luar untuk menghasilkan nilai bagi konsumen dan mendatangkan keuntungan. CRM berbasis pada data terkait konsumen yang berkualitas tinggi dan dilakukan oleh TI (Buttle, 2008). Menurut Francis Buttle (2008), beberapa perbedaan pendapat mengenai CRM dapat dijelaskan dengan mempertimbangkan adanya sejumlah tipe-tipe yang berbeda dalam mengidentifikasi CRM. Tipe-tipe tersebut adalah sebagai berikut :

1. *Strategic CRM*

Strategic CRM berfokus pada pengembangan budaya bisnis *customer-centric* yang bertujuan untuk memenangkan dan mempertahankan konsumen. Budaya bisnis *customer-centric* dapat dicapai dengan menciptakan dan memberikan nilai yang lebih baik dibandingkan dengan kompetitor. Dalam budaya seperti *customer-centric*, umumnya sumber daya yang ada akan ditempatkan pada posisi yang paling meningkatkan nilai konsumen, seperti menerapkan sistem *reward* untuk mendorong perilaku pegawai yang akan meningkatkan kepuasan pelanggan,

selain itu, informasi mengenai konsumen juga dikumpulkan, dibagikan dan diterapkan dalam bisnis.

2. *Operational CRM*

Operasional CRM mengotomatisasi dan mengembangkan proses-proses bisnis *customer-facing* dan *customer-supporting*. Software aplikasi CRM memungkinkan fungsi-fungsi *marketing*, penjualan dan servis yang telah terotomatisasi dan terintegrasi. Beberapa aplikasi yang umum digunakan dalam operasional CRM adalah :

a. *Marketing automation*

Marketing automation (MA) menggunakan teknologi sebagai proses pemasaran. Beberapa kemampuan ditawarkan oleh software MA : segmentasi pelanggan, manajemen kampanye dan pemasaran berbasis event. Perangkat lunak yang memungkinkan pengguna untuk mengeksplorasi data pelanggan mereka dalam rangka mengembangkan komunikasi yang ditargetkan dan ditawarkan. Segmentasi, dalam beberapa kasus, adalah mungkin pada tingkat pelanggan individu. Menawarkan unik dapat dibuat untuk seorang pelanggan pada titik waktu yang tepat .

b. *Sales-force automation*

Sales-force automation (SFA) adalah bentuk asli dari CRM. Di sini dipergunakan teknologi untuk melakukan manajemen perusahaan dalam menjual aktivitas. Proses penjualan dapat dibentuk dari berbagai tingkatan seperti menyediakan generasi, memberikan kualifikasi, identifikasi kebutuhan, pengembangan spesifikasi, membuat proposal, presentasi proposal, menangani obyek, dan menutup penjualan . Software Sales-force automation dapat dikonfigurasi sehingga hal ini dapat dibuat model untuk proses penjualan dari berbagai jenis industri dan organisasi .

c. *Service automation*

Service automation (SA) memungkinkan perusahaan untuk mengotomatisasi operasi layanan mereka, apakah disampaikan melalui *call center*, pusat kontak, web atau tatap muka di lapangan. Perangkat lunak yang memungkinkan perusahaan untuk mengelola dan mengkoordinasikan layanan terkait komunikasi *in-bound* dan *out-bound* di semua saluran.

3. *Analytical CRM*

Menurut Francis Buttle (2008) CRM *Analytical* atau *Analytical CRM* adalah proses mendapatkan, mengekstrak, memproses, dan melaporkan data yang terkait dengan pelanggan pada suatu perusahaan yang nantinya menjadi acuan dalam pengambilan sebuah keputusan. Data terkait pelanggan dapat ditemukan

di database sebuah perusahaan misal riwayat pembelian, data riwayat finansial, data pemasaran dan data pelayanan, serta data keluhan pelanggan (Buttle 2008). *Analytical CRM* juga berperan penting dalam menganalisis perilaku pelanggan. Data yang digunakan pada *Analytical CRM* adalah data yang berasal dari *CRM Operational* atau *Operational CRM*.

4. *Collaborative CRM*

Kolaborasi CRM merupakan istilah yang digunakan untuk mendeskripsikan arah strategi dan taktik pada perusahaan yang umumnya terpisah dalam *supply chain*, dengan tujuan keuntungan yang lebih dalam identifikasi, atraksi, *retention* dan pengembangan pelanggan. Kolaborasi CRM menggunakan teknologi CRM untuk berkomunikasi dan bertransaksi diseluruh bagian organisasi.

2.4 Manfaat CRM

Menurut Buttle (2008) manfaat yang dapat berpengaruh bagi meningkatnya nilai suatu perusahaan yaitu :

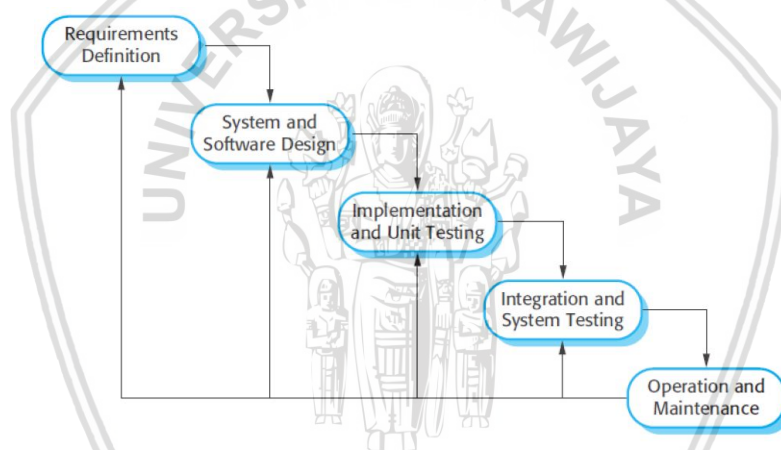
1. Meningkatkan Loyalitas Pelanggan Aplikasi CRM dengan menggunakan informasi dari semua titik dengan pelanggan, baik via web, *call center*, ataupun lewat staff pemasaran dan pelayanan di lapangan. Konsistensi dan aspek fleksibilitas informasi ini memungkinkan penjualan dan pelayanan yang lebih baik dengan berbagai informasi penting mengenai pelanggan tersebut.
2. Mengurangi Biaya, CRM juga memungkinkan penjualan atau pelayanan dengan biaya lebih murah dalam sebuah skema program pemasaran yang spesifik dan terfokus. Tertuju ke pelanggan yang tepat dan pada waktu yang tepat pula.
3. Meningkatkan Efisiensi Operasional, Otomasi penjualan dan proses layanan dari perusahaan dapat mengurangi resiko turunnya kualitas pelayanan dan mengurangi beban *cashflow*. Penggunaan teknologi web dan call center misalnya, akan mengurangi hambatan birokrasi dan biaya serta proses administratif yang mungkin timbul.
4. Peningkatan *Time to Market*, Aplikasi CRM membawa produk ke pasar dengan lebih cepat dengan informasi pelanggan yang lebih baik, adanya data trend pembelian oleh pelanggan.
5. Peningkatan Pendapatan, Aplikasi CRM menyediakan informasi untuk meningkatkan pendapatan dan keuntungan. Dengan aplikasi CRM, perusahaan dapat melakukan penjualan dan pelayanan melalui website sehingga peluang dari penjualan secara global tanpa perlu menyediakan upaya khusus untuk mendukung penjualan dan pelayanan tersebut.

2.5 Pengertian Sistem Informasi

Sistem adalah suatu proses yang saling berkaitan yang memiliki 3 aspek yakni input, proses dan output. Sebuah sistem dikatakan baik ketika sistem tersebut bisa memberikan output yang diharapkan dari input yang direncanakan. Sistem informasi terdiri dari gabungan teknologi, fasilitas, orang-orang, media, pengendalian dan prosedur yang bertujuan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberikan notifikasi kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan dasar informasi untuk pengambilan keputusan yang cerdas (Jogiyanto, 2005).

2.6 Metode Pengembangan Model *Waterfall*

Pengembangan model *waterfall* memiliki keruntutan dari satu fase ke fase lainnya, dengan demikian model ini dikenal dengan model *waterfall*. Pada model *waterfall* perancangan dilakukan dengan terencana terlebih dahulu karena digunakan sebagai dasar pengembangan kedepannya (Sommerville 2010).



Gambar 2.2 *Waterfall model*

Sumber: Sommerville (2010)

Pada Gambar 2.2 dapat dilihat metode pengembangan model *waterfall*. Tahapan utama dari model *waterfall* secara langsung menerapkan aktifitas dasar dalam pengembangan:

1. *Requirement analysis and definition* (Analisis kebutuhan dan definisi kebutuhan). Layanan, batasan, dan tujuan dibangun dengan berdiskusi dengan pengguna sistem. Lalu dijelaskan lebih rinci dan dijadikan sebagai spesifikasi sistem.
2. *System and Software design* (Sistem dan desain perangkat lunak). Pada desain sistem meliputi pendefinisian kebutuhan perangkat keras ataupun perangkat lunak untuk arsitektur sistem secara keseluruhan. Desain perangkat lunak meliputi identifikasi dan definisi fundamental dari sebuah abstraksi *software system* dan hubungan-hubungannya.

3. *Implementation and unit testing* (Implementasi dan pengujian unit). Pada tahap ini program direalisasikan dan diuji setiap unitnya untuk memastikan program yang dibuat sesuai dengan kebutuhan yang telah didefinisikan.
4. *Integration and System Testing*. Dari unit-unit yang diintegrasikan, digabungkan diuji, kemudian setelah program sudah lengkap secara keseluruhan sistem, program diujikan pada pengguna sistem.
5. *Operation and Maintenance*. Pada tahap ini sistem disintal dan digunakan. *Maintenance* meliputi perawatan *software*, membetulkan kesalahan-kesalahan atau *bug*, dan kekurangan-kekurangan pada program yang sebelumnya tidak dibahas pada tahap-tahap sebelumnya. Hal ini menyebabkan perulangan pada tahap sebelumnya untuk melakukan perubahan. Dalam penelitian ini tidak dilakukan tahap *operation and maintenance*.

2.7 Proses Bisnis

Proses bisnis merupakan sekumpulan aktivitas berelasi yang membutuhkan suatu *input* untuk menghasilkan *output* misal dalam bentuk laporan atau peramalan yang memiliki nilai bagi pelanggan. Proses bisnis juga memiliki definisi sebagai kumpulan aktivitas yang dirancang untuk menghasilkan *output* yang spesifik untuk pelanggan atau pasar tertentu. Proses bisnis memiliki komponen yaitu : (1) tujuan yang jelas, (2) adanya suatu masukan, (3) adanya suatu keluaran, (4) menggunakan sumber daya. (5) mempunyai kegiatan yang dibagi dalam beberapa tahap, (6) mempengaruhi lebih dari satu unit dalam organisasi dan (7) dapat menghasilkan nilai bagi konsumen (Monk, Ellen; Bret 2001).

2.8 Business Process Model Notation

Business Process Modelling Notation (BPMN) adalah bentuk pemodelan untuk menentukan bagaimana proses yang terjadi dalam alur bisnis dan siapa saja aktor yang terlibat dalam sebuah alur bisnis tersebut. BPMN digunakan untuk menyediakan notasi-notasi yang bisa dipahami oleh pengguna bisnis, dari analisis bisnis sebagai perancangan proses, orang yang mengimplementasikan teknologi, dan orang bisnis sebagai pengelola dan pengontrol proses tersebut (Bridgeland dan Zahavi 2009) .

Terdapat 4 kategori elemen dalam BPMN elemen sebagai penunjang dalam pembuatan BPMN (Bridgeland dan Zahavi 2009) . Kategori-kategori elemen tersebut adalah :

2.8.1 Flow Object

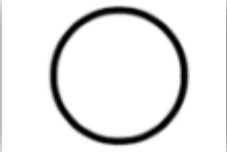
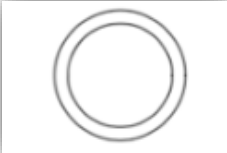

Terdapat 3 pembagian terhadap *flow object* yakni *event*, *activity*, dan *gateway*. Berikut penjelasannya :

1. Event

Event dapat didefinisikan sebagai suatu hal yang terjadi dan berdampak dalam proses bisnis. *Event* bisa berasal dari internal ataupun eksternal dari suatu proses. *Event* dibagi menjadi tiga yaitu *start event*, *intermediate event*, dan *end*

event. Pada setiap proses harus memiliki sebuah *start event* untuk sebagai awal dari proses bisnis. Pada Tabel 2.1 merupakan penjelasan dari tipe *event*.

Tabel 2.1 Tipe *event*

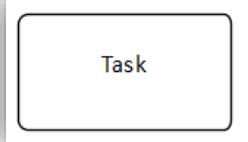
No	Tipe	Deskripsi	Simbol
1	<i>Start</i>	Proses dimulai.	
2	<i>Intermediate</i>	Terjadi diantara awal dan akhir proses. Berpengaruh pada alur dari proses, tapi tidak akan memulai atau menghentikan suatu proses.	
3	<i>End</i>	Akhir dari suatu proses.	

Sumber : (Bridgeland dan Zahavi 2009)

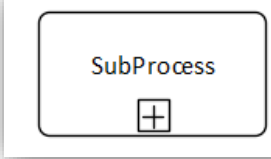
2. *Activity*

Activity dapat didefinisikan sebagai tugas yang dilakukan dalam proses bisnis. *Activity* digambarkan dengan kotak yang memiliki ujung bulat dan memiliki nama yang menjelaskan aktivitas yang dilakukan. Terdapat dua macam *activity* yaitu *task* dan *sub process*. Penjelasan dari tipe *activity* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tipe *activity*

No	Tipe	Deskripsi	Simbol
1	<i>Task</i>	Aktivitas yang dilakukan pada alur proses.	

Tabel 2.2 Tipe *activity*





No	Tipe	Deskripsi	Simbol
2	<i>Sub Process</i>	Aktivitas majemuk yang dimasukkan dalam proses. Aktivitas majemuk tersebut dapat dijelaskan dengan lebih rinci.	

Sumber : (Bridgeland and Zahavi 2009)

3. *Gateway*

Gateway memiliki peran dalam mengontrol alur dari sebuah proses bisnis. *Gateway* digambarkan dengan bentuk belah ketupat yang didalamnya terdapat simbol yang memiliki fungsi tertentu. Terdapat 4 jenis *gateway* yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tipe *gateway*

No	Tipe	Deskripsi	Simbol
1	<i>Exclusive</i>	<i>divergence</i> : untuk membuat jalur alternatif dalam sebuah proses, tetapi hanya salah satu yang dipilih. <i>convergence</i> : untuk mengabungkan jalur alternatif.	
2	<i>Parallel</i>	Proses yang dijalankan secara bersamaan.	
3	<i>Inclusive</i>	Sebuah proses yang dipecah menjadi beberapa jalur.	
4	<i>Complex</i>	Alur kompleks pada sebuah proses bisnis.	




Sumber : (Bridgeland and Zahavi 2009)

2.8.2 *Connections*

Connections dapat didefinisikan sebagai elemen penghubung *flow objects*. *Connections* memiliki 3 tipe yang masing masing memiliki bentuk

yang berbeda dan fungsi yang berbeda. Pada Tabel 2.4 merupakan penjelasan dari tipe *connections*.

Tabel 2.4 Tipe *connections*



No	Tipe	Deskripsi	Simbol
1	<i>Sequence Flow</i>	Menunjukkan urutan aktivitas yang dilakukan pada suatu proses.	
2	<i>Association</i>	Menunjukkan hubungan antara teks, <i>database</i> , artifak lain, dan <i>flow object</i> pada suatu proses.	
3	<i>Message Flow</i>	Menunjukkan alur pesan antara dua partisipan yang dapat mengirim dan menerima pesan.	

Sumber : (Bridgeland and Zahavi 2009)

2.8.3 *Swimlanes*

Swimlanes adalah wadah grafis yang mengelompokkan suatu set aktivitas dengan aktivitas lainnya. Terdapat 2 tipe swimlane yakni *pool* dan *lane*. Pada Tabel 2.5 merupakan penjelasan tipe *swimlanes*.

Tabel 2.5 Tipe *swimlanes*

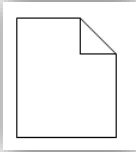



No	Tipe	Deskripsi	Simbol
1	<i>Pool</i>	Wadah yang berisikan satu proses dan <i>sequence flow</i> yang menjadi penghubung antar aktivitas.	
2	<i>Lane</i>	Mempresentasikan tanggungjawab aktivitas pada suatu proses.	

Sumber : (Bridgeland and Zahavi 2009)

2.8.4 Artifacts

Artifacts adalah objek luar dari sebuah proses. *Artifact* dapat berupa data atau catatan yang dapat digunakan untuk mengelola tugas atau proses. Terdapat 4 tipe *artifacts* yang akan dijelaskan pada Tabel 2.6.

Tabel 2.6 Tipe *artifacts*

No	Type	Deskripsi	Simbol
1	<i>Data Object</i>	Informasi yang mengalir pada sebuah proses misal dokumen bisnis, <i>email</i> , surat dan lain-lain.	
2	<i>Data Store</i>	Suatu tempat yang dapat digunakan dalam sebuah proses untuk membaca atau menulis data.	
3	<i>Annotation</i>	Menunjukkan informasi kepada pembaca berupa diagram BPMN.	
4	<i>Group</i>	Digunakan untuk mengelompokkan elemen secara informal.	

Sumber : (Bridgeland and Zahavi 2009)

2.9 Analisis Kebutuhan

Menurut Wazlawick (2014) ada dua tingkatan sudut pandang dalam melakukan analisis kebutuhan yakni *business perspective* dan *user perspective*.

2.9.1 Business Perspective

Business Perspective adalah sudut pandang yang melihat kebutuhan pengembangan aplikasi dari sisi fungsi dan proses bisnis untuk memenuhi tujuan bisnis yang ingin dicapai. Tujuannya adalah untuk memberikan pemahaman terhadap pemangku kepentingan bagaimana alur proses bisnis yang ada pada perusahaan tersebut yang bertujuan untuk mendapatkan gambaran secara jelas bagaimana proses bisnis yang sudah berjalan dan bagian mana dari proses bisnis tersebut yang akan ditingkatkan kinerjanya dengan menggunakan aplikasi tersebut sehingga mendapatkan kebutuhan yang mendukung tujuan bisnis yang

ingin dicapai. *Business Perspective* dapat disebut juga sebagai domain masalah (Monk and Bret J. Wagner 2013) .

2.9.1.1 Business Process (As-Is vs To-Be)

Business Process as-is menjelaskan secara keseluruhan proses bisnis yang berjalan saat ini di sebuah perusahaan atau organisasi dengan menggunakan pernyataan dan gambar. Tujuan dari analisis Business Process as-is untuk memperjelas dan memahami seperti apa kondisi saat ini dari proses bisnis yang ada dan bagaimana alur bisnis tersebut berjalan pada perusahaan atau organisasi (Monk, Ellen; Bret 2001).

Proses analisis yang dilakukan untuk mengetahui permasalahan dan kelemahan yang ada pada proses bisnis tersebut. Permasalahan tersebut dipetakan dalam tabel analisis permasalahan, sehingga bisa diketahui bagaimana solusi dan perbaikan untuk proses bisnis tersebut. Pada Tabel 2.7 merupakan cara dalam melakukan analisis permasalahan.

Tabel 2.7 Analisis permasalahan

Masalah	[Mendeskripsikan masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak dari masalah]
Solusi	[Solusi beserta manfaatnya]

Sumber : Bittner (2002)

Business Process to-be menjelaskan bagaimana peningkatan kinerja proses bisnis kedepannya kepada pemangku kepentingan perusahaan atau organisasi. Usulan yang diajukan berdasarkan apa yang sudah dilakukan pada analisis *Business Process as-is* untuk memenuhi seluruh permasalahan yang ada pada analisis *Business Process as-is*. Tujuan dari analisis *Business Process to-be* adalah untuk memberikan penjelasan dan gambaran kepada pemangku kepentingan perusahaan atau organisasi bagaimana usulan-usulan tersebut disatukan dengan proses bisnis yang ada, apakah sudah memenuhi keinginan dari pemangku kepentingan yang terlibat dalam proses bisnis tersebut (Monk, Ellen; Bret 2001)

2.9.1.2 Kebutuhan Pemangku Kepentingan

Menurut Bittner (2002) menjelaskan bahwa pemangku kepentingan atau pemangku kepentingan adalah seseorang yang mendapatkan manfaat secara langsung maupun tidak langsung dari sistem yang dikembangkan. Identifikasi kebutuhan dari pemangku kepentingan dilakukan untuk mengetahui bagaimana hubungan antara aktor dengan masalah-masalah yang ada sebelum diterapkannya aplikasi atau sistem. Dengan mengetahui siapa saja pemangku kepentingan yang akan menggunakan sistem tersebut, pengembang aplikasi bisa menentukan fungsi dan fitur apa saja yang ada dalam aplikasi dengan menyesuaikan karakteristik dari pemangku kepentingan tersebut, dimana pemangku kepentingan bertindak sebagai user atau pengguna dari sistem. Beberapa langkah dilakukan dalam analisis kebutuhan pemangku kepentingan adalah :

1. Tipe Pemangku Kepentingan

Tipe Pemangku kepentingan adalah klasifikasi sekelompok pemangku kepentingan yang berbagi karakteristik dan hubungan yang sama dengan sistem dan/atau proyek yang menghasilkan sistem. Menurut Bittner (2002), pemangku kepentingan biasanya masuk dalam kategori berikut:

- *Users*
Jenis pemangku kepentingan yang paling jelas adalah pengguna sebenarnya dari sistem. Mereka adalah orang-orang yang akan mengambil peran yang ditentukan oleh para aktor dalam model use-case.
- *Sponsors*
Para manajer bisnis, pemodal, pemegang saham, juara, kepala departemen, penjual, pemasar, anggota komite pengarah, dan orang lain yang berinvestasi dalam produksi sistem. Para pemangku kepentingan ini hanya pengguna tidak langsung dari sistem atau hanya dipengaruhi oleh hasil bisnis yang dipengaruhi oleh sistem. Banyak pembeli ekonomi untuk atau juara internal sistem.
- *Developers*
Manajer proyek, pengelola sistem, penguji, staf pendukung, perancang, pembuat kode, penulis teknis, staf produksi, dan setiap pengembang jenis lain yang terlibat dalam produksi dan dukungan sistem.
- *Authorities*
Para ahli dalam aspek tertentu dari masalah atau solusi domain. Ini termasuk otoritas legislatif, organisasi standar, departemen tata kelola organisasi, badan pengatur eksternal dan internal, pakar domain, dan pakar teknologi.
- *Customers*
Orang-orang dan / atau organisasi yang sebenarnya akan membeli sistem final. Ini dapat termasuk pembeli, evaluator, akuntan, dan agen yang bertindak atas nama organisasi pembelian.

Pada Tabel 2.8 merupakan cara dalam melakukan analisis tipe pemangku kepentingan.

Tabel 2.8 Analisis tipe pemangku kepentingan

Perwakilan Pemangku Kepentingan	Deskripsi	Tipe Pemangku Kepentingan
[Menyebutkan perwakilan pemangku kepentingan yang termasuk dalam tipe pemangku kepentingan]	[Menjelaskan representasi pemangku kepentingan terhadap sistem]	[Tipe pemangku kepentingan]

Sumber: Diadaptasi dari Bittner (2002)

2. Peran dan Perwakilan Pemangku Kepentingan

Peran pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki peran dan tanggung jawab yang sama terhadap pengembangan aplikasi. Hal tersebut bertujuan untuk mengetahui peran dan tanggung jawab masing-masing pemangku kepentingan sehingga menghindari kesalahan komunikasi pada saat proyek dilaksanakan.

Pemangku kepentingan sangat penting dalam proyek karena sistem yang dibuat digunakan secara langsung oleh pemangku kepentingan. Tabel 2.9 merupakan cara dalam melakukan analisis peran dan perwakilan pemangku kepentingan.

Tabel 2.9 Peran dan perwakilan pemangku kepentingan

Peran Pemangku Kepentingan	Perwakilan Pemangku Kepentingan	Tanggung Jawab	Kerlibatan
[Peran pemangku kepentingan]	[Perwakilan pemangku kepentingan]	[Menjelaskan tanggung jawab pemangku kepentingan]	[Menjaskan keterlibatan pemangku kepentingan terhadap sistem]

Sumber: Diadaptasi Bittner (2002)

3. Peran Pengguna

Peran pengguna merupakan peran yang dilakukan oleh pengguna saat berinteraksi dengan fungsi fungsi yang ada di dalam sistem. Peran pengguna juga dapat menggambarkan sistem lain yang melakukan interaksi dengan sistem.

4. Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan pemangku kepentingan menggambarkan masalah dalam proses bisnis atau pribadi dari pemangku kepentingan yang perlu dilakukan analisis bagaimana solusi yang dibutuhkan untuk mendukung tujuan bisnis dan menyelesaikan masalah yang ada. Identifikasi kebutuhan pemangku kepentingan dapat memberikan sebuah pandangan terhadap masalah utama dan solusi yang dapat menyelesaikan masalah tersebut (Bittner, 2002).

2.9.2 User Perspective

User Perspective adalah pandangan dari pengguna untuk membuat solusi untuk mengatasi masalah yang ada di *business perspective*.

2.9.2.1 Identifikasi Aktor

Identifikasi aktor menggambarkan pengguna yang benar-benar akan menggunakan sistem dan untuk siapa sistem memberikan nilai. Identifikasi aktor digunakan untuk menentukan hubungan antara jenis pengguna dan sistem yang

akan dibangun. Langkah pertama adalah menentukan peran yang akan digunakan oleh pengguna berhubungan dengan sistem. Peran terkait sistem ini dimodelkan sebagai aktor. Untuk setiap tipe pengguna yang diidentifikasi, setidaknya ada satu aktor yang didefinisikan untuk memenuhi kebutuhan mereka terhadap sistem (Bittner, 2002).

2.9.2.2 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem menggambarkan apa dan bagaimana sistem yang harus dilakukan oleh sebuah sistem. Kebutuhan-kebutuhan tersebut bergantung pada tipe aplikasi yang akan dikembangkan, apa yang diinginkan oleh pengguna dari aplikasi yang dikembangkan, dan bagaimana pendekatan dari pemangku kepentingan ketika menentukan kebutuhan aplikasi. Ketika menjelaskan kebutuhan dari pemangku kepentingan atau pengguna, Kebutuhan fungsional adalah salah satu cara untuk menjelaskan fungsi yang diberikan kepada pengguna yang bisa dipahami (Sommerville, 2010).

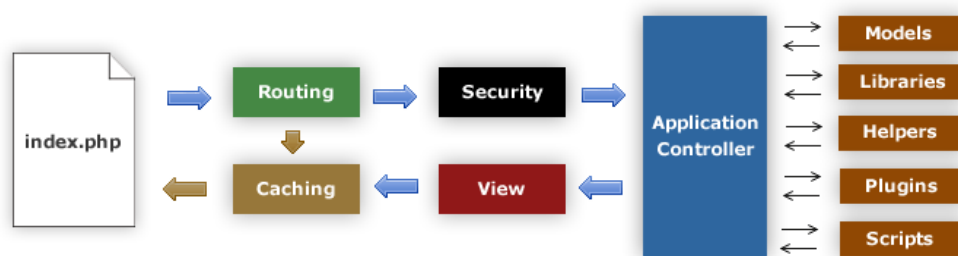
2.9.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak langsung berkaitan dengan layanan yang diberikan oleh sistem kepada pengguna. Kebutuhan ini berkaitan dengan keandalan sistem, waktu proses dari sistem, dan keamanan sistem. Kebutuhan non-fungsional juga menjelaskan bagaimana sebuah sistem dikembangkan sesuai dengan keadaan atau kondisi eksternal dari sistem seperti kebutuhan sistem untuk merespon perangkat *input output*, bagaimana sistem bisa menampilkan data yang representatif dan sesuai dengan keinginan pengguna, dan bagaimana sistem bisa menyesuaikan dengan sistem lain (Sommerville, 2010).

2.10 Code Igniter Framework

PHP adalah bahasa script yang umumnya digunakan untuk membuat aplikasi berbasis *web* yang pertama kali dikembangkan oleh Rasmus . *Framework* adalah kumpulan *script* php yang saling berkomunikasi dan bekerja bersama. *Code Igniter* merupakan aplikasi *open source* yang berupa framework dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP. *Code Igniter* memudahkan developer untuk membuat aplikasi *web* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *Code Igniter* dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir 1.6.3 yang dirilis pada 26 Juni 2008. *framework* ini menggunakan bahasa pemrograman php.

Code Igniter adalah *framework* php yang awalnya ditulis oleh Rick Ellis, pendiri dan CEO EllisLab.com perusahaan yang mengembangkan codeigniter. Saat ini, *Code Igniter* dikembangkan oleh komunitas dan disebarkan ke seluruh dunia dengan lisensi bebas. Pada Gambar 2.3 dapat dilihat arsitektur *Code Igniter Framework* .



Gambar 2.3 Arsitektur Code Igniter Framework

Sumber : (Sidik 2012)

2.11 Unified Model Language (UML)

UML adalah bahasa standar yang digunakan dan memvisualisasikan artefak dari proses analisis dan desain berorientasi objek. UML hanya merupakan bahasa pemodelan dan tidak menggambarkan bagaimana melakukan analisis dan desain berorientasi objek. Dalam pemodelan suatu sistem UML menyediakan standar pada notasi dan diagram. Pemodelan UML tidak didominasi oleh narasi, pemodelan dapat dilakukan secara visual. Pemodelan secara visual dapat membantu menangkap struktur dan perilaku dari objek, mempermudah penggambaran interaksi antar elemen dalam sistem, dan mempertahankan konsistensi antara desain dan implementasi dalam pemrograman (Rosa dan Shalahuddin, 2014). Menurut Rosa dan Shalahuddin (2013), ada dua tujuan penggunaan UML yaitu:

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

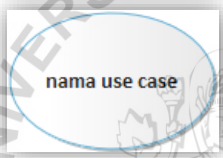


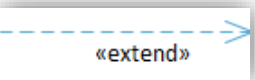
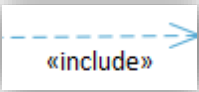
Secara konseptual, terdapat tiga unsur utama pada UML yaitu :

1. *Things*, pada UML terdapat empat jenis hal yaitu *Structural*, *Behavioral*, *Grouping*, dan *Annotation*.
2. *Relationship*, merupakan hubungan antar elemen pada sebuah pemodelan UML. Terdapat empat jenis relasi pada UML yaitu dependensi, asosiasi, generalisasi, dan realisasi.
3. *Diagram*, merupakan presentasi grafis dari serangkaian elemen yang sering diterjemahkan sebagai graf terhubung simpul (*things*) dan jalur (*relationship*). Terdapat 13 jenis diagram yang termasuk pada UML, diantaranya adalah *Class Diagram*, *Object Diagram*, *Component Diagram*, *Composite Structure Diagram*, *Use case diagram*, *Sequence diagram*, *Communication Diagram*, *State Diagram*, *Activity diagram*, *Deployment Diagram*, *Package Diagram*, *Timing Diagram*, dan *Interaction Overview Diagram*.


2.12 Use case diagram

Pengertian *Use case diagram* digunakan untuk menggambarkan interaksi antara pengguna dengan sistem (Rosa dan Shalahuddin, 2014). *Use case* menjelaskan apa saja yang dilakukan oleh aktor dan sistem tanpa menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case* adalah unit bagian dari fungsionalitas sistem yang menyediakan urutan pertukaran pesan oleh subjek dan satu atau lebih aktor dalam unit sistem. Sebuah *use case* melibatkan interaksi aktor dan sistem atau subjek lainnya (Rosa dan Shalahuddin, 2014). Simbol-simbol yang digunakan pada *use case diagram* dapat dilihat pada Tabel 2.10

Tabel 2.10 Simbol-simbol *use case diagram*

Nama	Simbol	Deskripsi
<i>Use case</i>		Spesifikasi tingkah laku sebuah entitas terhadap interaksi yang terjadi di luar.
Aktor / <i>actor</i>		Penggambaran peran dari pengguna di luar sistem, sistem atau sub sistem terhadap sistem. Setiap aktor berpartisipasi terhadap satu atau lebih <i>use case</i> di dalam sistem.
Asosiasi / <i>association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi / <i>extend</i>		Tingkah laku yang hanya berjalan jika ada kondisi tertentu.
Menggunakan / <i>include</i>		Tingkah laku yang harus dipenuhi supaya event dapat terjadi dan <i>use case</i> tersebut merupakan bagian dari <i>use case</i> lainnya

Tabel 2.10 Simbol-simbol *use case diagram*

Nama	Simbol	Deskripsi
Generalisasi / <i>generalization</i>		Hubungan antar <i>use case</i> yang umum dan lebih spesifik.


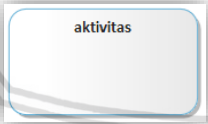

Sumber : (Rosa dan Shalahuddin, 2014)

Dengan menggunakan UML, *use case diagram* dapat diterapkan untuk menggambarkan perilaku sistem, sub sistem, atau *class* sehingga pengguna dapat memahami bagaimana dapat memahami bagaimana menggunakan elemen tersebut dan juga pengembang dapat mengimplementasikan elemen tersebut.

2.13 Activity diagram

Activity diagram adalah salah satu dari limas diagram di UML untuk memodelkan aspek dinamis dari sistem. *Activity diagram* pada umumnya merupakan sebuah *flowchart* yang menunjukkan aliran kontrol dari aktivitas ke aktivitas lainnya. *Activity diagram* menunjukkan *concurrency* serta percabangan (Rosa dan Shalahuddin, 2014). Tabel 2.11 merupakan keterangan dari simbol-simbol yang digunakan pada diagram aktivitas.

Tabel 2.11 Simbol-simbol *activity diagram*

Nama	Simbol	Deskripsi
Status awal		Status awal aktivitas dari sistem
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Branch /cabang		Sebuah cabang mungkin memiliki satu arus masuk dan dua atau lebih arus keluar dan digambarkan dalam simbol <i>diamond</i> . Pada setiap aliran keluar, ditempatkan ekspresi <i>Boolean</i> , yang dievaluasi saat memasuki cabang. Pada kondisi tersebut cabang berperan sebagai <i>decision</i> . Ketika dua aliran kontrol kembali bergabung bersama-sama, simbol <i>diamond</i> dapat digunakan dengan

Tabel 2.11 Simbol-simbol *activity diagram* (lanjutan)

Nama	Simbol	Deskripsi
		dua arus masuk dan satu arus keluar. Pada kondisi ini cabang berperan sebagai <i>merge</i> .
<i>Fork-join</i>		<i>Fork</i> merupakan pemisahan alur kontrol tunggal menjadi dua atau lebih alur kontrol bersamaan. Pada <i>fork</i> , mungkin memiliki satu transisi masuk dan dua atau lebih transisi keluar, masing-masing merepresentasikan aliran kontrol yang independen. <i>Join</i> merupakan sinkronisasi dua atau lebih alur kontrol yang bersamaan. Pada <i>join</i> mungkin memiliki dua atau lebih transisi masuk dan satu transisi keluar.
Status akhir		Status akhir yang dilakukan dari sistem
<i>Swimlane</i>		Bertujuan untuk partisi aktivitas pada diagram aktivitas dalam kelompok-kelompok, setiap kelompok mewakili organisasi bisnis yang bertanggung jawab.

Sumber : (Rosa dan Shalahuddin, 2014)


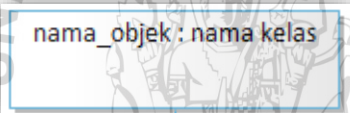


2.14 *Sequence diagram*

Sequence diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dll.) Dalam bentuk pesan atau pesan yang digambarkan terhadap waktu. *Sequence diagram* yang biasa menggambarkan skenario atau rangkaian langkah yang dilakukan dalam menanggapi suatu peristiwa untuk menghasilkan keluaran tertentu. Pesan yang dijelaskan dalam *sequence diagram* akan dipetakan ke fungsi / metode *class* pada

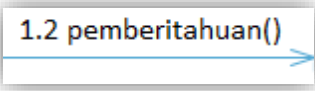
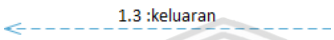
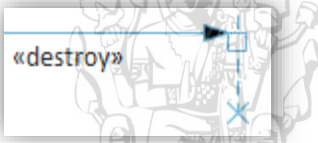
tahap perancangan selanjutnya. Untuk objek yang memiliki sifat khusus, standar UML mendefinisikan simbol khusus untuk objek *boundary*, *controller* dan *persistent entity* (Rosa dan Shalahuddin, 2014)

Sequence diagram merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. *Sequence diagram* digunakan untuk menggambarkan *dynamic view* dari sistem (Rosa dan Shalahuddin, 2014). Simbol-simbol yang digunakan pada *sequence diagram* ditunjukkan pada Tabel 2.12

Tabel 2.12 Simbol-simbol *sequence diagram*

Nama	Simbol	Deskripsi
<i>Lifeline</i>		Menunjukkan keberadaan objek selama periode waktu.
Objek		Menyatakan objek yang berinteraksi menggunakan pesan atau <i>message</i> .
<i>Activation Bar</i>		Menunjukkan periode waktu selama sebuah objek melakukan kegiatan, baik secara langsung maupun melalui prosedur perintah.
Pesan tipe <i>call</i> (<i>synchronous</i>)		Menyatakan suatu objek memanggil fungsi yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki fungsi, karena ini memanggil sebuah fungsi maka fungsi yang dipanggil harus ada pada <i>class diagram</i> sesuai

Tabel 2.12 Simbol-simbol *sequence diagram* (lanjutan)

Nama	Simbol	Deskripsi
		dengan <i>class</i> objek yang berinteraksi.
Pesan tipe <i>send</i> (<i>asynchronous</i>)		Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya
Pesan tipe <i>return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu fungsi menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i>		Objek dapat diakhiri selama interaksi yang berlangsung. <i>Lifeline</i> objek berakhir apabila menerima pesan <i>destroy</i> . Bila ada pesan tipe <i>create</i> maka sebaiknya ada pesan tipe <i>destroy</i> .

Sumber : (Rosa dan Shalahuddin, 2014)

2.15 Class Diagram

Class Diagram merupakan suatu spesifikasi, apabila spesifikasi tersebut diinstansiasi akan menghasilkan objek yang merupakan inti dari pengembangan berorientasi objek. *Class diagram* menggambarkan (atribut/properti) dari perangkat lunak, sekaligus berisi layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Rosa dan Shalahuddin, 2014).

Menurut Yasin suatu atribut dan fungsi suatu *class* dapat memiliki salah satu dari sifat berikut ini :

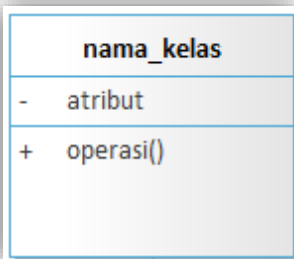

- Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
- Public*, dapat dipanggil oleh siapa saja.

Class diagram digunakan untuk memodelkan tampilan statis perancangan dari sebuah sistem. *Class diagram* tidak hanya penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural, tapi juga penting untuk menyusun sistem yang dapat dijalankan melalui teknik *forward* maupun *reverse*. *Class diagram* secara umum memuat tiga hal, yaitu *class*, *interfaces*, dan hubungan dependensi, generalisasi, serta asosiasi (Rosa dan Shalahuddin, 2014).





- Class*, adalah deskripsi dari sekumpulan objek yang memiliki atribut, operasi, hubungan, dan semantik yang sama. Sebuah *class* mengimplementasikan satu atau lebih *interfaces*.
- Interfaces*, dalam UML *interfaces* digunakan untuk memodelkan lapisan dalam suatu sistem yang merupakan kumpulan dari operasi yang digunakan untuk menentukan sebuah layanan dari *class* atau komponen. Dengan mendeklarasikan antarmuka maka dapat menyatakan perilaku dari independen abstraksi dari sebuah implementasi abstraksi tersebut.
- Dependensi, merupakan hubungan yang menyatakan bahwa sebuah hal (*class*) menggunakan informasi dan layanan dari hal lain (*class* lain), tetapi belum tentu sebaliknya. Biasanya menggunakan dependensi antar *class* untuk menunjukkan bahwa satu *class* menggunakan operasi dari *class* lain atau menggunakan variabel yang ada pada *class* lain.
- Generalisasi, adalah hubungan antara sebuah hal yang umum (disebut *superclass* atau *parent*) dan jenis hal yang lebih spesifik (disebut *subclass* atau *child*). Objek pada *child class* mungkin digunakan untuk variabel atau tipe parameter oleh *parent class*, tapi tidak sebaliknya. Hal ini berarti *child class* disubstitusikan untuk deklarasi *parent class*.
- Asosiasi, merupakan hubungan struktural yang menentukan bahwa objek dari satu *class* yang terhubung ke objek *class* lain.

Pada Tabel 2.13 menjelaskan tentang simbol-simbol yang digunakan dalam pembuatan *class diagram* pada umumnya.

Tabel 2.13 Simbol-simbol diagram *class*

Nama	Simbol	Deskripsi
<i>Class</i>		<i>Class</i> pada suatu struktur sistem.
Asosiasi / <i>association</i>		Relasi antar <i>class</i> dengan makna umum, asosiasi

Tabel 2.13 Simbol-simbol diagram *class*

Nama	Simbol	Deskripsi
		biasanya juga disertai dengan <i>multiplicity</i>
Ketergantungan / <i>dependency</i>		Ketergantungan antar <i>class</i> .
Agregasi / <i>aggregation</i>		Relasi antar <i>class</i> dengan makna semua-bagian (<i>whole-part</i>).
Asosiasi berarah / <i>directed association</i>		Relasi antar <i>class</i> dengan makna <i>class</i> yang satu digunakan oleh <i>class</i> yang lain.
Generalisasi		Relasi antar <i>class</i> dengan makna <i>class</i> yang satu digunakan oleh <i>class</i> yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

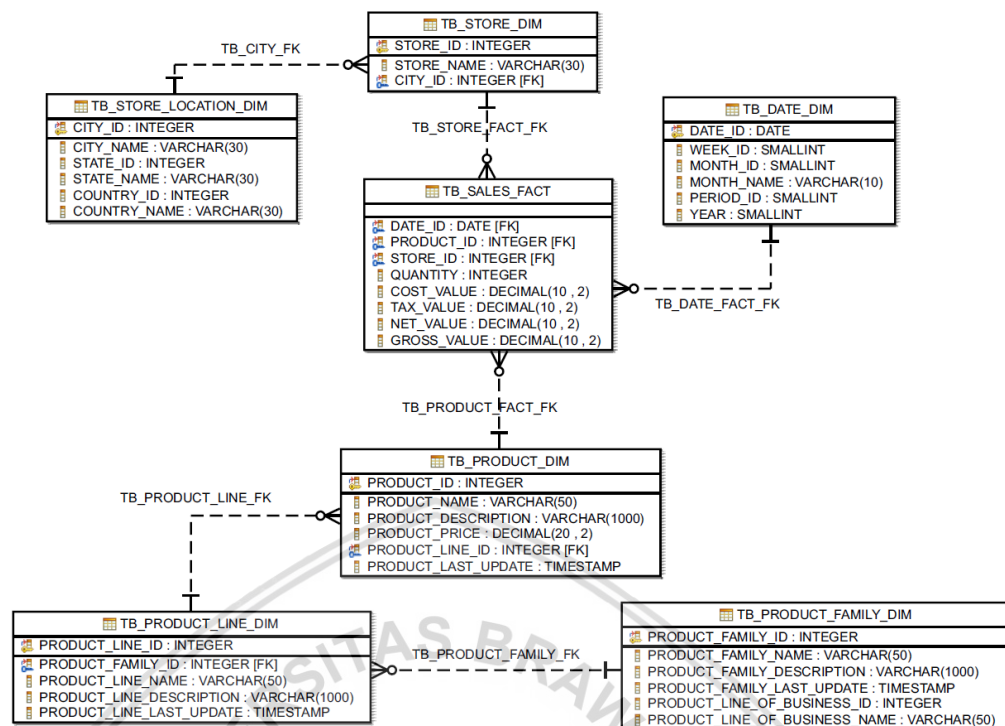
Sumber : (Rosa dan Shalahuddin, 2014)

2.16 Physical Data Model

Physical data model merepresentasikan bagaimana model akan dibangun di *database*. Sebuah *physical data model* menampilkan struktur tabel yang terdiri dari nama kolom, tipe data kolom, *constraints* kolom, *primary key*, *foreign key*, dan relasi antara tabel (IBM 2012). Terdapat beberapa tahap dalam pembuatan *physical data model*, yakni

1. Merubah entitas menjadi tabel.
2. Merubah relasi menjadi *foreign key*.
3. Merubah atribut menjadi kolom.
4. Memodifikasi *physical data model* sesuai dengan kebutuhan

Pada Gambar 2.4 dapat dilihat contoh *physical data model*.



Gambar 2.4 Physical data model

Sumber : (IBM 2012)

2.17 Pengujian White-Box

Pengujian *white-box testing* merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan perangkat komponen untuk menghasilkan *test-case* (Pressman, 2009).

Teknik yang digunakan dalam pengujian white-box menurut Tom McCabe adalah uji *basis path*. Metode ini memungkinkan perancang *test case* mendapatkan ukuran kekompleksan logika dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Test case yg didapat digunakan untuk mengerjakan basis set yg menjamin pengerjaan setiap perintah minimal satu kali selama uji coba (Pressman, 2009).

Cyclomatic complexity adalah metrik PL yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba *basis path*, nilai yang dihitung untuk *cyclomatic complexity* menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali (Pressman, 2009).

Dengan menggunakan pengujian *white-box* dapat memperoleh *test case* yang (1) menjamin bahwa semua jalur independen di dalam modul telah dieksekusi sedikitnya satu kali. (2) melaksanakan semua keputusan logis pada sisi benar dan

yang salah. (3) melaksanakan semua *loop* pada batas mereka dan dalam batas-batas operasional mereka, dan (4) melakukan struktur data internal untuk memastikan kesalahannya.

2.18 Pengujian *Black-Box*

Pengujian *black-box* berfokus pada kebutuhan fungsional dari sebuah perangkat lunak. Pada teknik pengujian ini akan ditentukan berbagai kondisi inputan yang akan dimasukkan pada program, dimana *input* tersebut akan memenuhi semua tugas (menguji) dari tiap fungsional program. Pengujian ini bermaksud untuk mengetahui apabila terdapat kesalahan dalam fungsi program, antarmuka, kesalahan pada data atau akses basis data, kesalahan performa serta kesalahan pada inisialisasi dan terminasi program (Pressman, 2009).

Black box testing menggunakan *validation testing* yang digunakan untuk menguji kebutuhan fungsional sedangkan untuk menguji kebutuhan non-fungsional akan menggunakan *compability testing*.

1. *Validation Testing*

Validation testing dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan. *Validation testing* termasuk ke dalam *black-box testing*, karena pengujian dilakukan tanpa menunjukkan proses berjalannya sebuah fungsi sebenarnya tetapi pengujian validasi melihat hasil yang dikeluarkan apakah sesuai dengan kebutuhan atau tidak dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan (Pressman, 2009).

2. *User Acceptance Testing*

User Acceptance Testing (UAT) merupakan pengujian yang menangani kebutuhan pengguna, *requirement*, dan *business process*. UAT dilakukan untuk mengetahui apakah suatu sistem telah memenuhi kriteria untuk diterima oleh pengguna atau *customer* (Pressman, 2009).

Menurut Pressman (2009) UAT biasanya dilakukan sebelum produk dihidupkan dan dilakukan setelah pengujian sistem. UAT dilakukan oleh *user* yang terlibat dengan sistem yang dibuat. Terdapat beberapa poin yang perlu diperhatikan dalam membuat UAT :

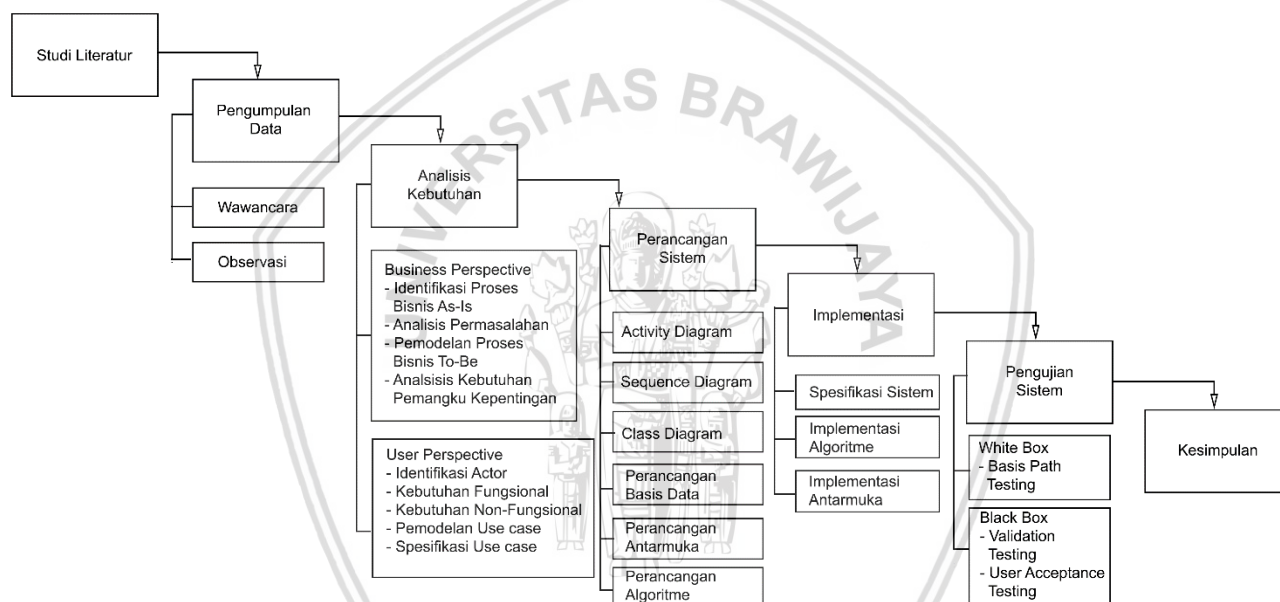
1. UAT bukan tentang halaman, *fields*, tombol. Asumsi tersebut dilakukan bahkan sebelum UAT dimulai. Semua hal mendasar itu seharusnya telah diuji dan bekerja dengan baik.
2. UAT adalah tentang entitas yang merupakan elemen utama dalam bisnis
3. UAT juga merupakan bentuk pengujian pada inti sistem yang berarti ada kemungkinan bagus untuk mengidentifikasi beberapa bug pada fase ini juga

4. UAT diklasifikasikan sebagai pengujian Alpha dan Beta, namun klasifikasi tersebut tidak begitu penting dalam konteks proyek pengembangan perangkat lunak biasa di industri berbasis layanan.
5. Sebagian besar waktu dalam proyek pengembangan perangkat lunak biasa, UAT dilakukan di lingkungan *quality assurance* jika tidak ada lingkungan pementasan atau UAT.



BAB 3 METODOLOGI

Metodologi penelitian yang digunakan dalam penelitian ini terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem dan kesimpulan. Metodologi penelitian yang digunakan dalam penelitian ini mengacu pada metode pengembangan *waterfall*. Metode pengembangan *waterfall* sering juga disebut model *sequential linear* atau *classic life cycle*. Tahapan pada metode pengembangan *waterfall* yang digunakan hanya meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian tidak sampai pada tahap *maintenance* dikarenakan penelitian ini bersifat implementatif. Setiap tahapan pada metodologi penelitian ini dilakukan secara runtut. Tahapan-tahapan metodologi penelitian yang diilustrasikan dapat dilihat pada diagram blok dalam Gambar 3.1



Gambar 3.1 Alur metode penelitian

3.1 Studi Literatur

Studi Literatur dilakukan dengan tujuan agar dapat menyelesaikan masalah dengan tepat dengan pengetahuan yang mendukung. Studi literatur dilakukan dengan mencari referensi-referensi yang relevan dengan objek yang akan diteliti. Pencarian referensi dilakukan di perpustakaan, maupun secara *online* melalui internet. Kemudian dilakukan pencarian data-data yang terkait dengan masalah pengaduan.

Setelah mendapatkan referensi-referensi yang relevan tersebut, dilakukan pencarian informasi-informasi yang dibutuhkan dalam penelitian ini dari referensi-referensi tersebut. Informasi yang didapatkan digunakan dalam penyusunan landasan kepastakaan, metodologi penelitian serta pengembangan sistem informasi. Pustaka-pustaka yang dijadikan acuan dapat dilihat di Daftar Pustaka.

3.2 Pengumpulan Data

Metode yang digunakan untuk pengumpulan data dalam penelitian ini adalah wawancara dan observasi.

3.2.1 Wawancara

Wawancara merupakan metode pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara ini merupakan salah satu metode pengumpulan data pada riset kualitatif. Namun, saat ini beberapa riset kuantitatif banyak juga yang menjadikan wawancara sebagai salah satu metode pengumpulan data. Dalam penelitian ini, wawancara dilakukan dengan memberikan sejumlah pertanyaan kepada pihak-pihak yang nantinya akan berhubungan dengan sistem yang akan dikembangkan ini.

3.2.2 Observasi

Pengumpulan data dilakukan dengan cara observasi di tempat penelitian. Observasi dilakukan dengan cara terjun langsung ke lapangan untuk mengetahui sistem yang sedang berjalan saat ini. Hal ini perlu dilakukan dengan tujuan untuk melakukan analisis terhadap sistem yang telah berjalan serta menentukan rancangan sistem baru yang akan dibangun agar tetap sinkron dengan sistem yang sudah ada.

3.3 Metode Pengembangan Sistem

Metode pengembangan yang digunakan pada penelitian ini adalah metode *waterfall*. Metode ini sudah digunakan secara luas untuk pengembangan aplikasi perangkat lunak dan merupakan model klasik dari rekayasa perangkat lunak yang banyak digunakan dalam proyek-proyek pemerintahan dan di perusahaan - perusahaan besar. Metode ini sangat terstruktur, tetapi cenderung bersifat linier dan tidak fleksibel. Metode ini memerlukan pendekatan yang sistematis dan sekuensial didalam pengembangan sistem perangkat lunaknya. Tahapan dari metode pengembangan *Waterfall* yang digunakan hanya terdiri dari analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.

3.3.1 Analisis Kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan user. Tahapan-tahapan di analisis kebutuhan ini antara lain sebagai berikut :

1. *Business Perspective*

Business Perspective digunakan untuk memperjelas dan memahami seperti apa kondisi penanganan dan pengelolaan pengaduan saat ini dari proses bisnis yang ada dan bagaimana alur bisnis tersebut berjalan. Tahapan ini dimulai dari pemodelan identifikasi proses bisnis *as-is*, analisis permasalahan, pemodelan proses bisnis *to-be*, analisis kebutuhan pemangku kepentingan. Identifikasi proses bisnis *as-is* digunakan untuk

menjelaskan sistem pengelolaan dan penanganan pengaduan yang berjalan saat ini dan digambarkan dengan notasi BPMN. Analisis permasalahan digunakan untuk menjelaskan lebih rinci tentang masalah pengaduan di kota XYZ yang sebelumnya dijelaskan di latar belakang dalam bentuk tabel analisis permasalahan. Kemudian dilakukan pemodelan proses bisnis *to-be* sebagai solusi terhadap masing-masing proses bisnis yang dibuat dalam diagram BPMN. Analisis kebutuhan pemangku kepentingan dilakukan untuk mendapatkan permasalahan apa saja yang dihadapi oleh pemangku kepentingan di kota XYZ dan solusi apa saja yang dibutuhkan untuk menyelesaikan permasalahan penanganan dan pengelolaan pengaduan sehingga target bisnis bisa tercapai.

2. *User Perspective*

User Perspective digunakan untuk membuat solusi untuk mengatasi masalah yang ada di *business perspective*. Tahapan ini dimulai dari identifikasi aktor, kebutuhan fungsional, kebutuhan non-fungsional. Identifikasi aktor digunakan untuk menjelaskan mengenai siapa saja aktor dalam sistem *E-Complaint*. Kemudian dilakukan pengelompokan kebutuhan yang berdasarkan analisis kebutuhan pemangku kepentingan ke dalam kebutuhan fungsional dan kebutuhan non-fungsional.

3. *Use case diagram*

Use case diagram digunakan untuk menggambarkan sistem *E-Complaint* dari sudut pandang pengguna sistem (*user*), sehingga pembuatan *use case diagram* ini lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

3.3.2 Perancangan Sistem

Pada Tahap perancangan sistem digunakan diagram-diagram untuk menggambarkan kerja sistem. Perancangan dalam penelitian ini menggunakan *Unified Modelling Language* (UML). Diagram-diagram UML yang dibuat antara lain sebagai berikut :

1. *Activity diagram*

Activity diagram digunakan untuk menggambarkan rangkaian aliran dari aktivitas dari sistem *E-Complaint*. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Penggambaran *activity diagram* di bagian ini hanya pada beberapa fitur saja.

2. *Sequence diagram*

Sequence diagram digunakan untuk menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem *E-Complaint*. Penggambaran *sequence diagram* pada bagian ini hanya menggunakan beberapa fitur saja.

3. *Class diagram*

Class diagram digunakan untuk mendeskripsikan jenis – jenis obyek dalam sistem *E-Complaint* dan berbagai macam hubungan statis yang terjadi. *Class diagram* dibagi menjadi dua yaitu *class diagram* untuk implementasi dan *class diagram* sebagai domain model. *Class diagram* untuk implementasi digunakan untuk perancangan sistem *E-Complaint* dan *class diagram* domain model digunakan untuk perancangan basis data.

4. Perancangan Basis Data

Perancangan basis data digunakan untuk merancang *database* yang akan digunakan di sistem *E-Complaint*. Perancangan basis data dibuat dengan mendefinisikan struktur-struktur pada tabel yang terdapat pada *class diagram* domain model. Perancangan basis data secara logika diwujudkan dalam bentuk *Entity Relational Diagram* (ERD) konseptual. Setelah itu dilakukan proses pembuatan *Physical Data Model* (PDM).

5. Perancangan Antarmuka

Selanjutnya dilakukan perancangan antarmuka (interface), perancangan antarmuka ini dilakukan untuk merancang tata letak sistem *E-Complaint* sesuai dengan analisis kebutuhan sistem. Perancangan antarmuka pada bagian ini hanya menggunakan beberapa halaman saja.

6. Perancangan Algoritme

Perancangan Algoritme digunakan untuk menjelaskan algoritme yang dipakai untuk diimplementasikan di sistem *E-Complaint*. Perancangan Algoritme pada bagian ini hanya beberapa fitur saja.

3.3.3 Implementasi

Pada bagian ini dilakukan implementasi dari rancangan yang sudah dibuat ke dalam baris-baris kode dengan menggunakan bahasa PHP. Dalam implementasi *web* ini menggunakan *framework Code Igniter* untuk menunjang proses pengembangan sesuai dengan kebutuhan. Tahapan-tahapan di implementasi ini antara lain sebagai berikut :

1. Spesifikasi Sistem

Spesifikasi Sistem digunakan untuk menjelaskan mengenai spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem *E-Complaint*.

2. Implementasi *Class Diagram*

Implementasi *class diagram* dilakukan dengan mengubah kelas-kelas pada *class diagram* mengubah menjadi file-file php dan mengubah atribut beserta metodenya menjadi *source code*.

3. Implmentasi Basis Data

Implementasi basis data dilakukan dengan mengubah PDM ke dalam *source code* dengan bahasa mysql.

4. Implementasi Algoritme

Implementasi Algoritme dilakukan dengan mengubah algoritme yang dibuat di tahap perancangan algoritme ke dalam kode program. Implementasi pada bagian ini hanya pada fitur utama saja.

5. Implementasi Antarmuka

Implementasi Antarmuka dilakukan dengan cara mengimplementasikan perancangan antarmuka pada tahap sebelumnya melalui kode program pada sistem *E-Complaint*. Implementasi antarmuka pada bagian ini hanya pada halaman utama saja.

3.3.4 Pengujian

Pada tahap pengujian dilakukan pengujian-pengujian untuk memastikan sistem *E-Complaint* memiliki kelayakan dan sesuai dengan kebutuhan. Pada tahap ini terdapat pengujian sebagai berikut :

1. Pengujian *White-Box*

Pengujian dengan metode *white-box* dilakukan dengan menggunakan pengujian basis path dimana pengujian dilakukan untuk mendapatkan kompleksitas logika dari sistem *E-Complaint* dan menggunakannya sebagai petunjuk untuk mendefinisikan semua jalur pengerjaan menggunakan pengujian *basis path testing*. Pengujian basis path pada bagian ini hanya menggunakan dua kasus pengujian.

2. Pengujian *Black-Box*

Pengujian *black-box* ini bertujuan untuk memastikan bahwa setiap kebutuhan fungsional yang telah didefinisikan sebelumnya mampu dipenuhi oleh sistem *E-Complaint*. Pengujian menggunakan *validation testing* dan *user acceptance testing*. Pengujian *validasi* dilakukan untuk mengetahui apakah sistem *E-Complaint* berjalan sesuai dengan kebutuhan yang telah didefinisikan. Pengujian validasi pada bagian ini hanya menggunakan beberapa fitur utama saja. Pengujian UAT dilakukan untuk mengetahui apakah sistem *E-Complaint* telah memenuhi kriteria untuk diterima oleh pengguna. Pengujian UAT dilakukan dengan memberikan lembar persetujuan berisikan pertanyaan-pertanyaan yang diajukan kepada *stakeholder*.

3.3.5 Pengambilan Kesimpulan dan Saran

Penarikan kesimpulan dilakukan sebagai hasil dari penelitian ini dimana sistem yang dikembangkan telah memenuhi kebutuhan pengguna dan memecahkan masalah yang telah dirumuskan sebelumnya.

BAB 4 ANALISIS KEBUTUHAN

Bab ini akan membahas tentang analisis kebutuhan terhadap sistem yang akan dibangun. Terdapat juga sub bab yang menjelaskan tentang penjelasan proses bisnis, pemaparan masalah, penentuan pemangku kepentingan, identifikasi kebutuhan berdasarkan pemangku kepentingan, kebutuhan fungsional dan non-fungsional, identifikasi aktor, pemodelan *use case* dan spesifikasi *use case*.

4.1 Business Perspective

Di bagian *Business Perspective* akan dilakukan analisis terhadap sistem yang digunakan dalam penelitian ini. Hasil dari analisis ini didapatkan alur proses bisnis, pemangku kepentingan, dan permasalahan yang dihadapi oleh pemangku kepentingan.

4.1.1 Identifikasi Proses Bisnis AS-IS

Berdasarkan hasil wawancara yang dilakukan, diperoleh bagaimana proses pengaduan berjalan saat ini.

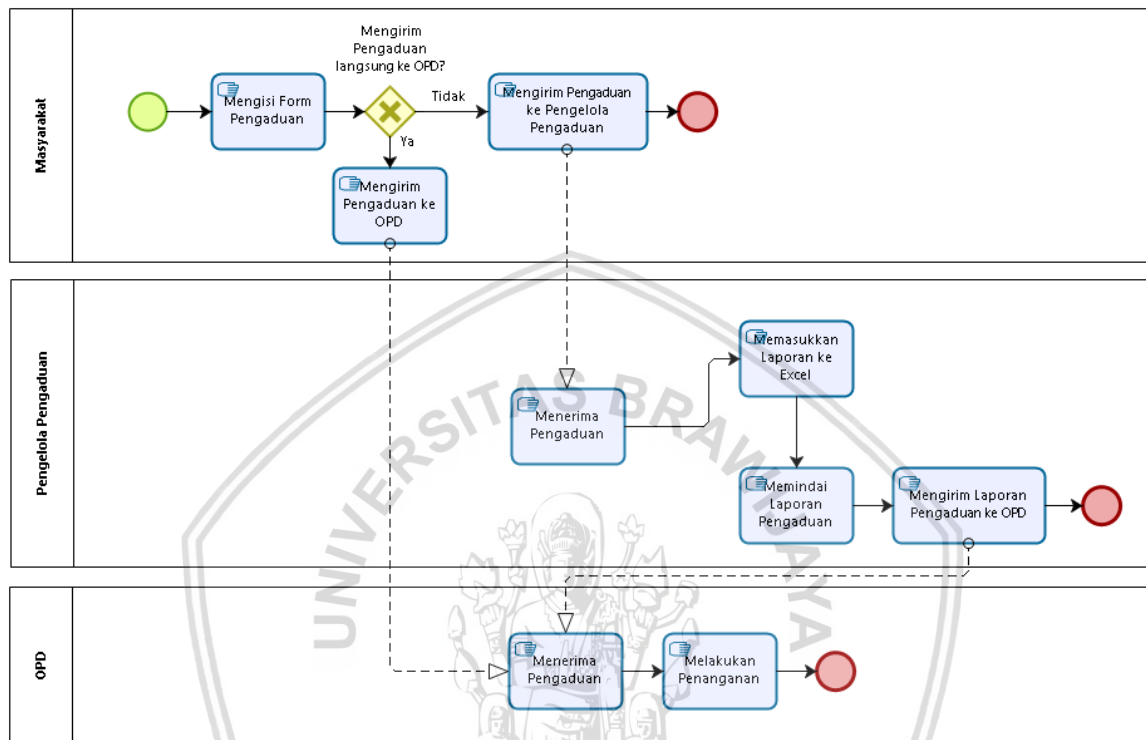
4.1.1.1 Proses Pengelolaan dan Penanganan Pengaduan

Pada sistem pengelolaan dan penanganan pengaduan yang berjalan saat ini, masyarakat dapat melaporkan pengaduan melalui sms, media sosial atau melaporkan secara langsung ke pihak pelayanan pengaduan atau ke OPD di kota XYZ. Untuk pelaporan melalui pihak pelayanan pengaduan masyarakat diharuskan mengisi formulir identitas dan menyampaikan permasalahan pengaduan. Jika pengaduan yang dilaporkan perlu ditindaklanjuti oleh OPD lain maka pihak pelayanan pengaduan akan menyampaikan permasalahan ke OPD bersangkutan dengan mengirimkan hasil pindai laporan pengaduan untuk mendapatkan penanganan pengaduan. Untuk proses pencaataan pengaduan, pengaduan yang masuk hanya direkap pada suatu buku sehingga pihak pengelola pengaduan sulit untuk melakukan klasifikasi dan kategorisasi. Untuk proses klasifikasi dan kategorisasi pengaduan harus dimasukkan ke Microsoft Excel terlebih dahulu. Untuk proses pembuatan laporan terkait data pengaduan di kota XYZ pihak pengelola pengaduan harus mengelola lagi data yang telah dimasukkan ke Microsoft Excel sesuai dengan laporan yang dibutuhkan.

Pada sistem pengaduan yang berjalan saat ini terdapat beberapa permasalahan yaitu terdapat kemungkinan formulir pengaduan tidak terisi secara lengkap sehingga pelaksanaan penanganan pengaduan membutuhkan waktu yang lebih lama dari seharusnya. Permasalahan berikutnya masyarakat kesulitan untuk mengetahui tindak lanjut dari pengaduan yang mereka kirim karena mereka harus menghubungi pihak pelayanan pengaduan. Seharusnya laporan pengaduan bisa dikelola dengan baik secara terpusat dan terintegrasi serta dapat diakses secara *real time* sehingga bisa memudahkan pihak-pihak yang membutuhkan informasi tentang pengaduan di kota XYZ dan bisa dijadikan acuan dalam pengambilan keputusan di kota XYZ. Proses pelaporan pengaduan seharusnya dapat

memberikan kepuasan untuk masyarakat seperti sistem dapat memberikan *feedback* dari laporan yang mereka kirim dan kemudahan untuk mengetahui pengaduan mereka sudah ditindaklanjuti atau tidak.

Pada Gambar 4.1 adalah penggambaran notasi BPMN untuk menggambarkan bagaimana proses bisnis pengelolaan dan penanganan pengaduan sistem lama.



Gambar 4.1 Diagram BPMN proses bisnis pengelolaan dan penanganan

4.1.2 Analisis Permasalahan

Bagian analisis permasalahan akan menjelaskan lebih rinci tentang masalah yang sebelumnya dijelaskan di proses bisnis *as-is* dalam bentuk tabel analisis permasalahan yang ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Analisis permasalahan

Masalah	<ol style="list-style-type: none"> 1. Sistem pengaduan yang berjalan saat ini tidak memiliki pemusatan data dikarenakan masyarakat dapat mengirimkan data pengaduan melalui OPD tertentu saja dan pengelola pengaduan di kota XYZ. 2. Sistem saat ini tidak dapat memantau langsung apakah OPD di kota XYZ melakukan penanganan pengaduan dengan benar di lapangan atau tidak. 3. Pengelolaan data pengaduan masih dilakukan dengan cara merekap laporan pengaduan yang masuk ke dalam sebuah buku dan kemudian data tersebut akan dimasukkan lagi ke Microsoft Excel untuk dikelola menjadi laporan.
----------------	--

Tabel 4.1 Analisis permasalahan

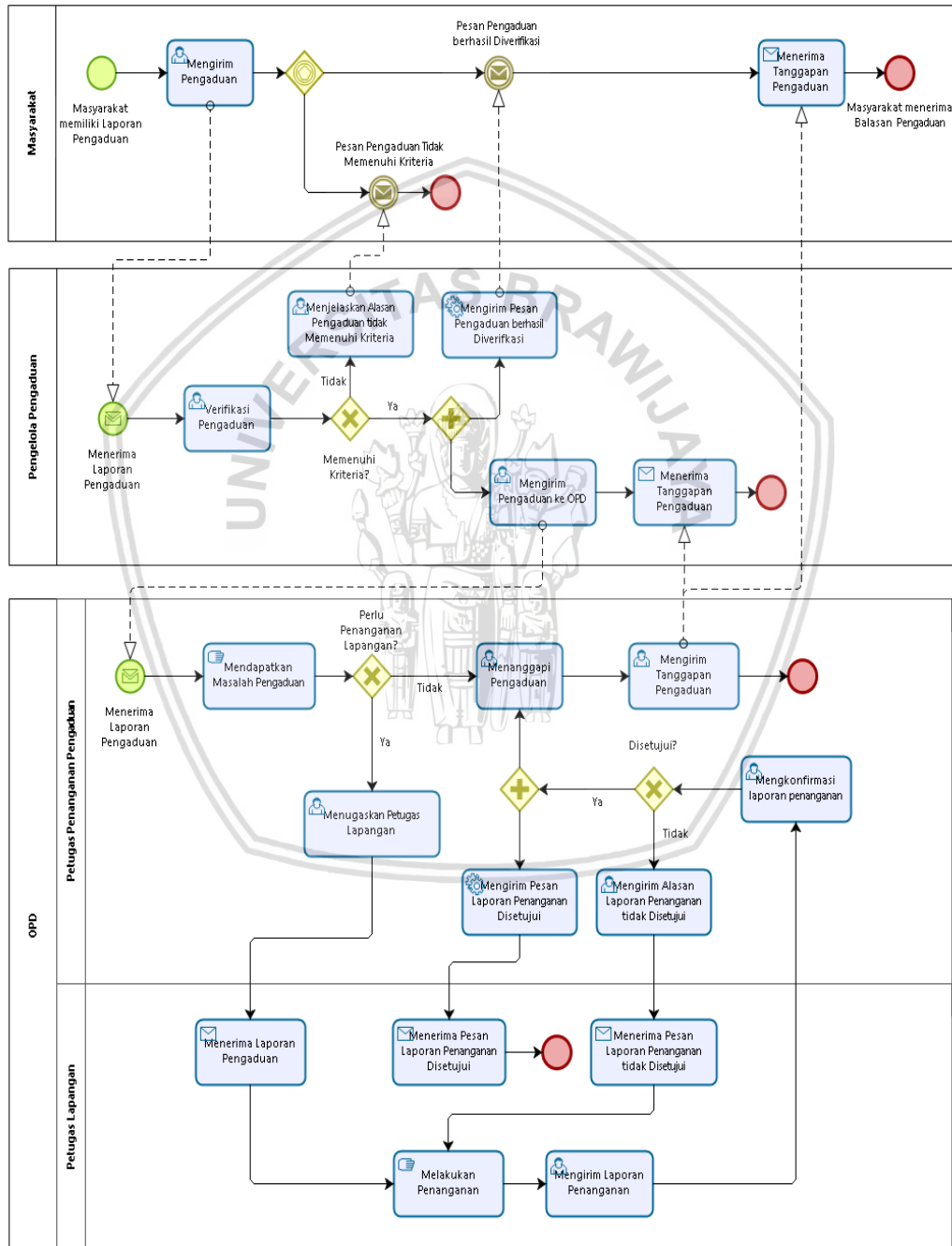
	<ol style="list-style-type: none"> 4. Pelaporan pengaduan ke OPD terkait masih menggunakan dengan mengirimkan hasil pindai laporan pengaduan. 5. Sistem saat ini tidak dapat memberikan <i>feedback</i> kepada masyarakat terkait dengan pengaduan yang mereka kirim. 6. Sistem pengaduan saat ini tidak saling terintegrasi antar OPD maupun pihak-pihak pengelola pengaduan di kota XYZ. 7. Sistem saat ini tidak dapat melakukan pelaporan secara <i>real time</i> untuk semua pihak dikarenakan pengaduan proses pengelolaan pengaduan membutuhkan waktu yang lama. 8. Membutuhkan waktu yang lebih lama untuk proses pembuatan laporan terkait data pengaduan di kota XYZ
Mempengaruhi	Masyarakat, Pihak Pengelola Pengaduan, OPD yang menerima pengaduan.
Dampak	<ol style="list-style-type: none"> 1. Kesulitan untuk menemukan data terkait pengaduan di kota XYZ dan kesulitan menghubungi masyarakat. 2. Terdapat kemungkinan pengaduan tidak dilakukan penanganan. 3. Dapat menyebabkan peluang kehilangan data pengaduan dari masyarakat. 4. Terjadi pemborosan tenaga kerja untuk proses pengiriman pengaduan ke OPD lain. 5. Ketidakpuasan masyarakat terhadap kinerja pengelola pengaduan dikarenakan tidak dapat mengetahui apakah pengaduan mereka sudah ditindaklanjuti atau tidak. 6. Kesulitan untuk menghubungi pihak-pihak yang bertugas melakukan penanganan pengaduan. 7. Membutuhkan waktu yang lebih lama untuk proses penanganan pengaduan. 8. Proses pembuatan laporan menjadi tidak efisien.
Solusi	Membangun sebuah sistem informasi yang dapat mengelola pengaduan dengan baik secara terpusat dan terintegrasi serta dapat diakses secara <i>real time</i> sehingga bisa memudahkan pihak-pihak yang membutuhkan informasi tentang pengaduan di kota XYZ dan bisa dijadikan acuan dalam pengambilan keputusan di kota XYZ. Dapat memberikan kepuasan untuk masyarakat seperti sistem dapat memberikan <i>feedback</i> dari laporan yang mereka kirim dan kemudahan untuk mengetahui pengaduan mereka sudah ditindaklanjuti atau tidak.

4.1.3 Pemodelan Proses Bisnis To-Be

Dari hasil analisis permasalahan yang diperoleh, didapatkan solusi terhadap masing-masing proses bisnis yang dibuat dalam diagram BPMN.

4.1.3.1 Proses Pengelolaan dan Penanganan Pengaduan

Proses pengelolaan dan penanganan pengaduan yang sudah diperbarui memiliki alur proses yang digambarkan pada Gambar 4.2.



Gambar 4.2 Diagram BPMN proses bisnis pengelolaan dan penanganan pengaduan sistem baru

Pada sistem pengelolaan dan penanganan pengaduan sistem baru, masyarakat dapat melaporkan pengaduan melalui *web* dengan mengisi *form* yang sudah disediakan. Kemudian masyarakat dapat mengirimkan pengaduan mereka secara langsung melalui *web*. Setelah itu pengaduan akan diterima oleh pihak pengelola pengaduan dan dilakukan proses verifikasi. Proses verifikasi dilakukan untuk mengetahui apakah pengaduan sudah sesuai dengan ruang lingkup pelayanan pengaduan atau tidak, isi pengaduan sudah jelas atau lengkap dan laporan pengaduan tidak termasuk *spam* dan mengandung unsur SARA.

Jika verifikasi memenuhi kriteria pengaduan maka pengaduan akan dikirim ke OPD sesuai dengan permasalahan pengaduan dan sistem akan memberikan pesan kepada masyarakat “pengaduan berhasil diverifikasi”. Jika verifikasi tidak memenuhi kriteria maka pengelola pengaduan akan memberikan penjelasan mengapa pengaduan tidak memenuhi kriteria ke masyarakat. Setelah pengaduan diterima oleh Petugas Penanganan Pengaduan, pengaduan akan diperiksa apakah pengaduan perlu dilakukan penanganan di lapangan atau tidak. Jika tidak maka Petugas Penanganan Pengaduan dapat menanggapi langsung pengaduan dari masyarakat. Jika diperlukan penanganan lapangan maka Petugas Penanganan Pengaduan akan menugaskan Petugas Lapangan untuk melakukan penanganan di lapangan dengan cara mengirim laporan pengaduan ke Petugas Lapangan. Setelah Petugas Lapangan melakukan penanganan di lapangan, Petugas Lapangan akan mengirimkan laporan penanganan ke Petugas Penanganan Pengaduan untuk dikonfirmasi.

Jika laporan penanganan tidak disetujui maka Petugas Penanganan Pengaduan akan mengirimkan alasan mengapa laporan penanganan tidak disetujui. Kemudian petugas lapangan melakukan penanganan di lapangan dan mengirim laporan penanganan. Jika laporan penanganan pengaduan disetujui maka Petugas Penanganan Pengaduan dapat menanggapi laporan pengaduan dan dapat mengirim tanggapan pengaduan ke masyarakat dan pengelola pengaduan.

Pada alur proses yang baru terdapat perubahan-perubahan sebagai berikut:

1. Masyarakat dapat mengirimkan pengaduannya melalui *web E-Complaint* dengan cara mengisi formulir yang sudah disediakan.
2. Pengelola Pengaduan dapat memverifikasi pengaduan untuk menentukan pengaduan memenuhi kriteria atau tidak.
3. Pengelola Pengaduan dapat mengirim pengaduan ke OPD sesuai dengan permasalahan pengaduan tersebut melalui sistem.
4. Petugas Penanganan Pengaduan dapat melakukan konfirmasi laporan pengaduan perlu dilakukan penanganan di lapangan atau tidak.
5. Petugas Penanganan Pengaduan dapat membalas pengaduan secara langsung pengaduan jika tidak diperlukan penanganan.
6. Petugas Penanganan Pengaduan dapat menugaskan petugas lapangan untuk melakukan penanganan di lapangan dengan mengirim laporan pengaduan ke Petugas Lapangan jika diperlukan penanganan di lapangan.

7. Petugas Lapangan dapat mengirim laporan penanganan melalui sistem.
8. Petugas Penanganan Pengaduan dapat mengkonfirmasi laporan pengaduan sudah ditangani atau belum.
9. Petugas Penanganan Pengaduan dapat mengirim tanggapan pengaduan ke masyarakat sebagai balasan pengaduan.
10. Masyarakat dapat menerima balasan pengaduan dari pengaduan yang mereka kirim.

4.1.4 Analisis Kebutuhan Pemangku Kepentingan

Analisis kebutuhan pemangku kepentingan dilakukan untuk mendapatkan permasalahan apa saja yang dihadapi oleh pemangku kepentingan dan solusi apa saja yang dibutuhkan untuk menyelesaikan permasalahan tersebut sehingga target bisnis bisa tercapai. Langkah-langkah yang dilakukan sebagai berikut :

4.1.4.1 Identifikasi Pemangku Kepentingan dan Tipe Pemangku Kepentingan

Bagian ini akan menjelaskan tentang tipe pemangku kepentingan dan perwakilan pemangku kepentingan yang berhubungan dengan sistem. Tipe tipe tersebut dijelaskan pada Tabel 4.2.

Tabel 4.2 Pemangku Kepentingan dan Tipe Pemangku Kepentingan

Perwakilan Pemangku Kepentingan	Deskripsi	Tipe Pemangku Kepentingan
Pengelola Pengaduan, Petugas Penanganan Pengaduan, Petugas Lapangan	Pengguna sistem yang menggunakan fungsi sistem untuk melakukan pengelolaan dan penanganan pengaduan	<i>Users</i>
Masyarakat	Pengguna sistem yang menggunakan sistem untuk melaporkan pengaduan	<i>Users</i>
Kepala Unit Pengelola, Kepala OPD	Pengguna sistem yang menggunakan fungsi sistem untuk memantau/mengawasi pengelolaan dan penanganan pengaduan	<i>Users</i>

4.1.4.2 Identifikasi Peran dari Perwakilan Pemangku Kepentingan

Bagian ini akan menjelaskan tentang peran dari perwakilan pemangku kepentingan yang berhubungan dengan sistem. Tipe tipe tersebut dijelaskan pada Tabel 4.3.

Tabel 4.3 Peran dari Perwakilan Pemangku Kepentingan

Peran Pemangku Kepentingan	Perwakilan Pemangku Kepentingan	Tanggung Jawab	Keterlibatan
Pengguna Umum	Masyarakat	Bertanggung jawab melaporkan pengaduan di kota XYZ	Mengirim pengaduan, mencari data pengaduan
Pengelola Pengaduan	Pengelola Pengaduan	Bertanggung jawab mengelola pengaduan di kota XYZ	Memverifikasi pengaduan, menyalurkan pengaduan ke OPD
Petugas Penanganan	Petugas Penanganan Pengaduan, Petugas Lapangan	Bertanggung jawab untuk melakukan penanganan pengaduan dan memberikan tanggapan pengaduan di kota XYZ	Menanggapi pengaduan, mendelegasikan petugas lapangan, mengkonfirmasi laporan penanganan, melakukan penanganan di lapangan, mengirim laporan penanganan
Pengawas Pengaduan	Kepala Unit Pengelola, Kepala OPD	Bertanggung jawab untuk mengawasi/memonitor kinerja pengelolaan dan penanganan pengaduan di kota XYZ	Mengawasi pengelolaan dan penanganan pengaduan di kota XYZ

4.1.4.3 Identifikasi Kebutuhan Pemangku Kepentingan

Tahapan ini menjelaskan tentang kebutuhan yang harus dipenuhi pemangku kepentingan dan alasan mengapa kebutuhan tersebut harus dipenuhi. Terdapat Kebutuhan, pemangku kepentingan, situasi saat ini, dan solusi yang ditawarkan. Pada kolom Pemangku kepentingan, situasi saat ini, dan solusi yang ditawarkan digunakan untuk memetakan setiap kebutuhan berasal darimana dan hasil analisis yang didapatkan. Kebutuhan tersebut dijelaskan pada Tabel 4.4.

Tabel 4.4 Kebutuhan pemangku kepentingan

No	Kebutuhan	Perwakilan Pemangku kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
1	Memberikan kemudahan kepada masyarakat untuk melaporkan pengaduan dan dapat menerima <i>feedback</i>	Masyarakat	Saat ini masyarakat hanya bisa melaporkan pengaduan melalui sms, media sosial atau melaporkan secara langsung ke pihak pelayanan pengaduan dan sistem saat ini tidak bisa memberikan <i>feedback</i> kepada masyarakat	Terdapat fitur mengirim pengaduan untuk masyarakat melalui <i>web</i> dan fitur untuk melihat status pengaduan sudah diproses atau belum
2	Memberikan kemudahan bagi pihak pengelola pengaduan memverifikasi dan menyalurkan pengaduan ke OPD di Kota XYZ	Pengelola pengaduan	Penyaluran laporan ke OPD di kota XYZ dilakukan dengan cara mengirim hasil pindai laporan pengaduan	Terdapat fitur untuk memverifikasi pengaduan dan mengirim laporan pengaduan ke OPD di kota XYZ
3	Memberikan kemudahan dalam mengelola laporan pengaduan di kota XYZ	Pengelola pengaduan	Proses pencatatan laporan pengaduan dilakukan dengan cara mencatat setiap laporan pengaduan ke Microsoft Excel	Sistem dapat menyimpan setiap laporan pengaduan secara otomatis
4	Memberikan kemudahan dalam menampung pengaduan di kota XYZ untuk	Pengelola pengaduan	Masyarakat dapat mengirim pengaduan mereka langsung ke OPD tertentu sehingga pihak pengelola pengaduan kesulitan dalam memperoleh data	Sistem pengaduan dibuat secara terpusat sehingga pihak pengelola pengaduan dapat menerima setiap laporan pengaduan di kota XYZ

Tabel 4.4 Kebutuhan pemangku kepentingan

No	Kebutuhan	Perwakilan Pemangku kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
			pengaduan tersebut	
5	Memberikan kemudahan untuk menerima tanggapan pengaduan dan menyampaikan tanggapan pengaduan ke masyarakat	Pengelola pengaduan	Pihak pengelola pengaduan kesulitan membalas laporan pengaduan ke masyarakat dikarenakan pihak pengelola pengaduan kesulitan untuk mendapatkan laporan penanganan dari OPD di kota XYZ	Terdapat fitur untuk menerima tanggapan pengaduan dari OPD di kota XYZ dan menyampaikan tanggapan pengaduan ke masyarakat
6	Memberikan kemudahan bagi Petugas Penanganan Pengaduan untuk menanggapi laporan pengaduan	Petugas Penanganan Pengaduan	Petugas penanganan pengaduan kesulitan menyampaikan hasil penanganan pengaduan ke masyarakat	Terdapat fitur untuk menanggapi laporan pengaduan dari masyarakat
7	Mendelegasikan Petugas Lapangan untuk melakukan penanganan pengaduan di lapangan	Petugas Penanganan Pengaduan	Penugasan Petugas Lapangan masih dilakukan dengan membuat laporan penugasan	Terdapat fitur untuk menugaskan petugas lapangan untuk melakukan penanganan pengaduan
8	Mengkonfirmasi laporan pengaduan sudah ditangani atau belum.	Petugas Penanganan Pengaduan	Konfirmasi laporan pengaduan dilakukan dengan cara menyetujui laporan penanganan yang	Terdapat fitur untuk mengkonfirmasi laporan penanganan dari petugas lapangan

Tabel 4.4 Kebutuhan pemangku kepentingan

No	Kebutuhan	Perwakilan Pemangku kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
			dibuat oleh petugas lapangan	
9	Mengawasi/memantau kinerja pengelolaan dan penanganan pengaduan di kota XYZ	Kepala Unit Pengelola, Kepala OPD	Laporan pengaduan dan penanganan pengaduan tidak dapat diakses secara <i>real time</i> dan tidak terintegrasi antar OPD	Terdapat fitur untuk melihat dashboard terkait data pengaduan dan penanganan pengaduan di kota XYZ
10	Membuat laporan terkait data pengaduan di kota XYZ	Pengelola pengaduan, Petugas Penanganan pengaduan	Pembuatan laporan dilakukan dengan mengelola kembali data yang ada di excel sesuai dengan laporan yang dibutuhkan	Terdapat fitur untuk membuat laporan terkait data pengaduan dengan format excel
11	Keamanan	Masyarakat, Pihak pengelola pengaduan, OPD di kota XYZ	-	Terdapat proses autentifikasi pada sistem yang dibangun
12	Memberikan kemudahan akses bagi pengguna sistem	Masyarakat, Pihak pengelola pengaduan, OPD di kota XYZ	Saat ini belum ada sistem pengaduan berbasis <i>web</i> untuk pengguna sistem	Sistem dapat diakses melalui <i>web</i> selama terdapat koneksi internet dan memiliki <i>browser</i> pada <i>device</i> yang digunakan

4.2 User Prespective

User Perspective dilakukan untuk menganalisis kebutuhan sistem yang digunakan dalam penelitian ini. Hasil dari analisis ini didapatkan identifikasi aktor, kebutuhan fungsional, kebutuhan non-fungsional, *use case diagram* dan spesifikasi *use case*.

4.2.1 Identifikasi Aktor

Tahapan ini akan menjelaskan mengenai siapa saja aktor dalam sistem berdasarkan peran pemangku kepentingan pada tabel 4.3 . Penjelasan dari aktor dapat dilihat pada tabel 4.5.

Tabel 4.5 Identifikasi aktor

Aktor	Tipe Pengguna	Deskripsi
Pengunjung	<i>Standard Users</i>	Pengguna umum yang membutuhkan akses <i>login</i> atau daftar dan mencari laporan pengaduan di kota XYZ
Masyarakat	<i>Standard Users</i>	Pengguna umum yang menggunakan sistem untuk mendukung aktifitas pelaporan pengaduan di kota XYZ.
Pengelola Pengaduan	<i>Technology Adopters</i>	Pihak pengelola pengaduan yang bertanggung jawab untuk memverifikasi pengaduan dan mengirim pengaduan ke Admin OPD di kota XYZ
Petugas Penanganan Pengaduan	<i>Technology Adopters</i>	Staff OPD yang bertanggung jawab untuk menanggapi laporan pengaduan dan mendelegasikan petugas lapangan untuk melakukan penanganan di lapangan
Petugas Lapangan	<i>Technology Adopters</i>	Staff OPD yang bertanggung jawab untuk menangani pengaduan di lapangan dan melaporkan hasil penanganan
Kepala Unit Pengelola	<i>Technology Adopters</i>	Pengguna yang membutuhkan data pengaduan di kota XYZ.
Kepala OPD	<i>Technology Adopters</i>	Pengguna yang membutuhkan data pengaduan di OPD XYZ.

4.2.2 Kebutuhan Fungsional dan Non-Fungsional

Tahapan ini akan menjelaskan mengenai kebutuhan-kebutuhan yang harus dipenuhi oleh sistem secara deskriptif. Kebutuhan-kebutuhan tersebut dibagi menjadi dua macam, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan pengelompokan kebutuhan yang harus disediakan oleh sistem secara fungsional didalamnya. Kebutuhan ini didasarkan

pada kebutuhan pemangku kepentingan dan pengguna yang terdapat di bagian sebelumnya. Kebutuhan fungsional tersebut dijelaskan pada Tabel 4.6.

Tabel 4.6 Kebutuhan Fungsional

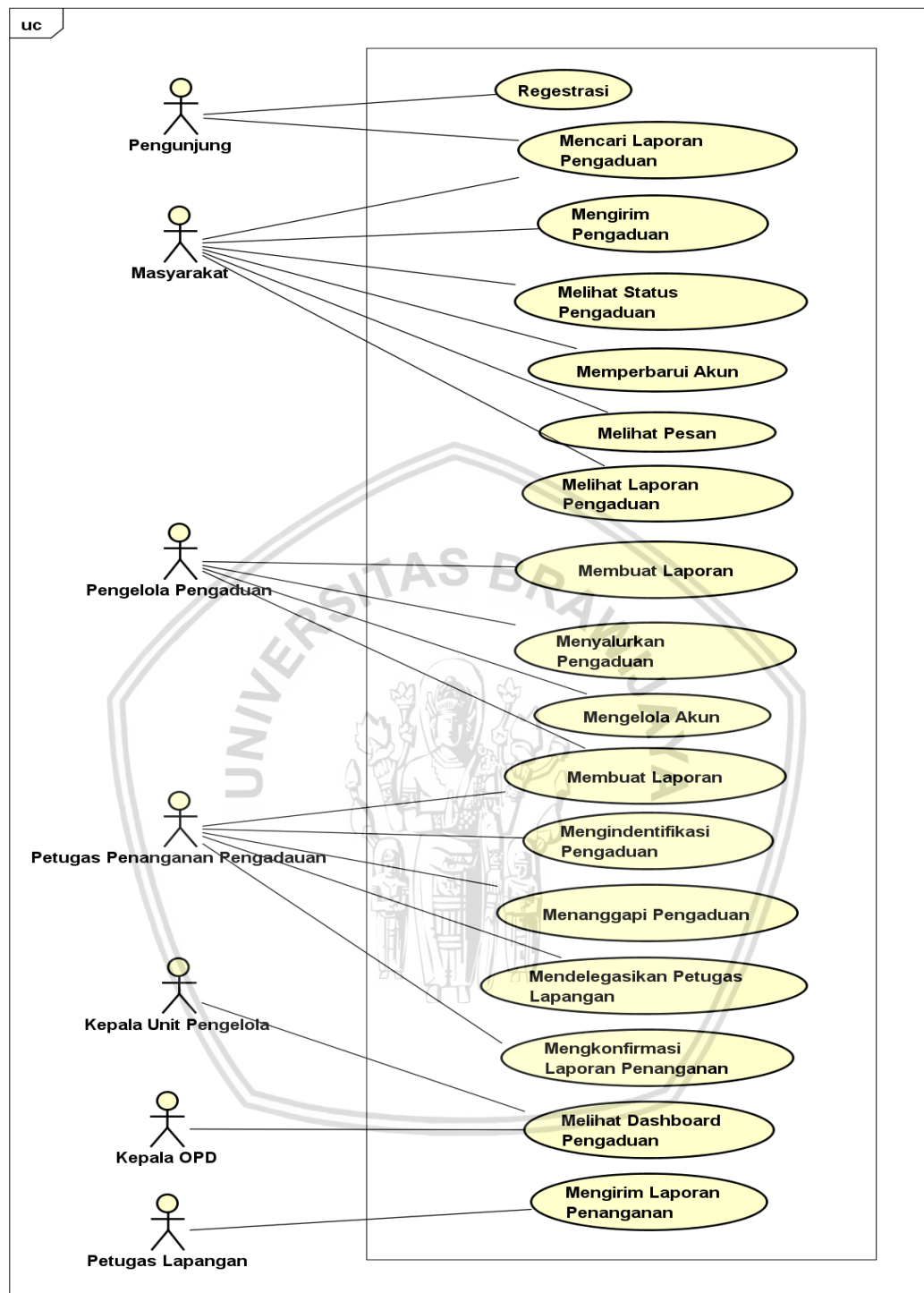
Kode Fungsi	Nama Fungsi	Deskripsi
SEC-FT-1	Mengirim Pengaduan	Sistem dapat digunakan mengirim laporan pengaduan
SEC-FT-2	Verifikasi Pengaduan	Sistem dapat digunakan untuk memverifikasi pengaduan sudah memenuhi kriteria atau tidak
SEC-FT-3	Menyalurkan Pengaduan	Sistem dapat digunakan untuk menyalurkan pengaduan dari masyarakat ke OPD di kota XYZ
SEC-FT-4	Mengidentifikasi Pengaduan	Sistem dapat digunakan untuk mengkonfirmasi laporan pengaduan perlu dilakukan penanganan pengaduan di lapangan atau tidak
SEC-FT-5	Menanggapi Pengaduan	Sistem dapat digunakan untuk menanggapi laporan pengaduan dari masyarakat
SEC-FT-6	Mendelegasikan Petugas Lapangan	Sistem dapat digunakan untuk mendelegasikan Petugas Lapangan untuk melakukan penanganan pengaduan di lapangan
SEC-FT-7	Mengirim Laporan Penanganan	Sistem dapat digunakan untuk mengirim laporan penanganan ke Petugas Penanganan Pengaduan
SEC-FT-8	Mengkonfirmasi Laporan Penanganan	Sistem dapat digunakan untuk mengkonfirmasi laporan pengaduan sudah ditangani atau belum
SEC-FT-9	Melihat Pesan	Sistem dapat digunakan untuk menampilkan pesan
SEC-FT-10	Melihat Status Pengaduan	Sistem dapat digunakan untuk melihat status pengaduan sedang di proses atau sudah ditangani
SEC-FT-11	Mencari Laporan Pengaduan	Sistem dapat digunakan untuk mencari laporan pengaduan
SEC-FT-12	Melihat <i>Dashboard</i> Pengaduan	Sistem dapat menampilkan jumlah laporan pengaduan perbulan
SEC-FT-13	Melihat Laporan Pengaduan	Sistem dapat menampilkan laporan pengaduan untuk semua <i>user</i>

Tabel 4.6 Kebutuhan Fungsional (lanjutan)

Kode Fungsi	Nama Fungsi	Deskripsi
SEC-FT-14	Registrasi	Sistem dapat digunakan untuk mendaftar sebagai pengguna <i>E-Complaint</i>
SEC-FT-15	Mengelola akun	Sistem dapat digunakan untuk mengolah akun masyarakat yang sudah terdaftar dan mengolah akun di OPD XYZ
SEC-FT-16	<i>Login</i>	Sistem dapat digunakan untuk melakukan validasi untuk masuk ke dalam sistem
SEC-FT-17	<i>Logout</i>	Sistem dapat digunakan untuk keluar dari sistem
SEC-FT-18	Memperbarui akun	Sistem dapat digunakan untuk melakukan pengelolaan akun untuk merubah <i>password</i> dan memperbarui identitas

4.3 Use Case Diagram

Pada tahap ini digambarkan sistem dalam bentuk *use case diagram* sesuai kebutuhan fungsional yang telah dibuat sebelumnya pada Tabel 4.6. Kemudian masing-masing *use case* dijelaskan lebih spesifik ke dalam tabel spesifikasi *use case*.



Gambar 4.3 Use case diagram

4.4 Spesifikasi Use Case

Setiap kebutuhan yang ada pada *use case diagram* akan dijelaskan lebih rinci pada bagian spesifikasi *use case*. Tabel spesifikasi *use case* terdiri dari nama *use case*, deskripsi dari *use case* terkait, aktor yang menggunakannya, *pre-conditions*,

skenario utama dan skenario alternatif yang dilakukan ketika menggunakan *use case* tersebut dan *post-conditions*.

4.4.1 Spesifikasi *Use case* Mengirim Pengaduan

Tabel 4.7 Spesifikasi *use case* mengirim pengaduan

Use case name	Mengirim pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor mengirim pengaduan
Actor	Masyarakat
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu mengirim pengaduan 2. Sistem menampilkan <i>form</i> pengaduan 3. Aktor mengisi <i>form</i> pengaduan 4. Aktor memilih untuk mengirim pengaduan 5. Sistem menyimpan data pengaduan 6. <i>Use case</i> selesai.
Alternative Flows	<p>Exception 4: kesalahan pengisian data <i>form</i> tidak lengkap</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan pesan <i>error</i> pada <i>form</i> yang belum dilengkapi atau terdapat kesalahan pengisian. 2. Kembali ke langkah 3
Post-Conditions	Laporan pengaduan disimpan ke <i>database</i> dan sistem menampilkan notifikasi “pengaduan berhasil dikirim”.

4.4.2 Spesifikasi *Use case* Verifikasi Pengaduan

Tabel 4.8 Spesifikasi *use case* verifikasi pengaduan

Use case name	Verifikasi pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor memverifikasi pengaduan
Actor	Pengelola Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu pengaduan. 2. Sistem menampilkan daftar pengaduan. 3. Aktor memilih operasi verifikasi pengaduan <ul style="list-style-type: none"> • Memenuhi Kriteria: variant 3a • Tidak Memenuhi Kriteria : variant 3b

Tabel 4.8 Spesifikasi *use case* verifikasi pengaduan

	4. <i>Use case</i> selesai.
Alternative Flows	<p>Variant 3a: Memenuhi Kriteria</p> <ol style="list-style-type: none"> 1. Aktor memilih memenuhi kriteria 2. Sistem mengirimkan pesan ke masyarakat "Pengaduan sudah dikirim ke OPD" 3. Sistem merubah status pengaduan menjadi "berhasil diverifikasi" 4. <i>Use case</i> selesai <p>Variant 3b: Tidak Memenuhi Kriteria</p> <ol style="list-style-type: none"> 1. Aktor memilih tidak memenuhi kriteria 2. Sistem menampilkan <i>form</i> pesan 3. Aktor mengisi <i>form</i> pesan 4. Aktor memilih untuk mengirim pesan 5. Sistem mengirimkan pesan ke masyarakat 6. <i>Use case</i> selesai
Post-Conditions	Status pengaduan berubah menjadi "sudah diverifikasi" dan sistem menampilkan notifikasi "Pengaduan berhasil diverifikasi".

4.4.3 Spesifikasi *Use case* Menyalurkan Pengaduan

Tabel 4.9 Spesifikasi *use case* menyalurkan pengaduan

Use case name	Menyalurkan pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor menyalurkan pengaduan
Actor	Pengelola Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan pengaduan sudah berhasil diverifikasi.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu pengaduan. 2. Sistem menampilkan daftar pengaduan. 3. Aktor memilih menu kirim pengaduan. 4. Sistem menampilkan list OPD di kota XYZ. 5. Aktor memilih OPD di kota XYZ 6. Aktor memilih untuk mengirim pengaduan. 7. Sistem merubah status pengaduan menjadi "pengaduan sudah dikirim ke OPD" 8. <i>Use case</i> selesai.
Alternative Flows	<p>Exception 6: kesalahan aktor tidak memilih OPD</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan pesan <i>error</i> "Anda belum memilih OPD".

Tabel 4.9 Spesifikasi *use case* menyalurkan pengaduan

	2. Kembali ke langkah 5
Post-Conditions	Status pengaduan berubah menjadi “pengaduan telah dikirim ke OPD” Sistem menampilkan notifikasi “Pengaduan berhasil dikirim ke OPD”.

4.4.4 Spesifikasi *Use case* Mengidentifikasi Pengaduan

Tabel 4.10 Spesifikasi *use case* mengidentifikasi pengaduan

Use case name	Mengidentifikasi pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor mengidentifikasi pengaduan
Actor	Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Aktor memilih menu pengaduan. 2. Sistem menampilkan daftar pengaduan. 3. Aktor memilih menu konfirmasi pengaduan. 4. Sistem menampilkan konfirmasi “Apakah pengaduan perlu dilakukan penanganan lapangan atau tidak”. 5. Aktor memilih konfirmasi : <ul style="list-style-type: none"> • Ya : variant 5a • Tidak : variant 5b 6. <i>Use case</i> selesai.
Alternative Flows	<p>Variant 5a: Ya</p> <ol style="list-style-type: none"> 1. Aktor memilih konfirmasi “Ya” 2. Sistem merubah status pengaduan menjadi “pengaduan masih dilakukan penanganan” 3. <i>Use case</i> selesai <p>Variant 5b: Tidak</p> <ol style="list-style-type: none"> 1. Aktor memilih konfirmasi “Tidak” 2. Sistem menampilkan menu tanggap pengaduan 3. <i>Use case</i> selesai
Post-Conditions	Status pengaduan berubah menjadi “pengaduan perlu dilakukan penanganan lapangan” dan sistem menampilkan notifikasi “pengaduan berhasil diidentifikasi”.

4.4.5 Spesifikasi Use case Mendelegasikan Petugas Lapangan

Tabel 4.11 Spesifikasi use case mendelegasikan petugas lapangan

Use case name	Mendelegasikan Petugas Lapangan
Brief Description	Use case ini mendeskripsikan tentang bagaimana aktor mendelegasikan petugas lapangan
Actor	Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Use case ini bermula ketika aktor memilih menu pengaduan. 2. Sistem menampilkan daftar pengaduan. 3. Aktor memilih menu kirim pengaduan. 4. Sistem menampilkan <i>list</i> petugas lapangan di OPD XYZ. 5. Aktor memilih petugas lapangan yang tersedia. 6. Aktor memilih untuk mengirim pengaduan. 7. Sistem merubah status pengaduan menjadi “pengaduan sedang dilakukan penanganan” 8. Use case selesai.
Alternative Flows	<p>Exception 6: kesalahan pengisian data <i>form</i> tidak diisi</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan pesan <i>error</i> pada <i>form</i> yang belum dilengkapi atau terdapat kesalahan pengisian. 2. Kembali ke langkah 5
Post-Conditions	Status pengaduan berubah menjadi “pengaduan sedang dilakukan penanganan lapangan” dan sistem menampilkan notifikasi “pengaduan berhasil dikonfirmasi”.

4.4.6 Spesifikasi Use case Mengirim Laporan Penanganan

Tabel 4.12 Spesifikasi use case mengirim laporan penanganan

Use case name	Mengirim laporan penanganan
Brief Description	Use case ini mendeskripsikan tentang bagaimana aktor mengirim laporan penanganan
Actor	Petugas Lapangan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Use case ini bermula ketika aktor memilih menu mengirim laporan 2. Sistem menampilkan <i>form</i> laporan penanganan 3. Aktor mengisi <i>form</i> laporan penanganan

Tabel 4.12 Spesifikasi *use case* mengirim laporan penanganan

	<ol style="list-style-type: none"> Aktor memilih untuk mengirim laporan penanganan Sistem menyimpan data laporan penanganan <i>Use case</i> selesai.
Alternative Flows	<p>Exception 4: kesalahan pengisian data <i>form</i> tidak lengkap</p> <ol style="list-style-type: none"> Sistem akan menampilkan pesan <i>error</i> pada <i>form</i> yang belum dilengkapi atau terdapat kesalahan pengisian. Kembali ke langkah 3
Post-Conditions	Laporan penanganan disimpan ke <i>database</i> dan sistem menampilkan notifikasi “laporan penanganan berhasil dikirim”.

4.4.7 Spesifikasi *Use case* Mengkonfirmasi Laporan Penanganan

Tabel 4.13 Spesifikasi *use case* mengkonfirmasi laporan penanganan

Use case name	Mengkonfirmasi laporan penanganan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor mengkonfirmasi laporan penanganan
Actor	Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> Aktor memilih menu laporan penanganan. Sistem menampilkan daftar laporan penanganan. Aktor memilih menu konfirmasi laporan penanganan. Sistem menampilkan konfirmasi “Apakah pengaduan sudah dilakukan penanganan atau tidak”. Aktor memilih konfirmasi : <ul style="list-style-type: none"> Ya : variant 5a Tidak : variant 5b <i>Use case</i> selesai.
Alternative Flows	<p>Variant 5a: Ya</p> <ol style="list-style-type: none"> Aktor memilih konfirmasi “Ya” Sistem mengirimkan pesan ke petugas lapangan “Laporan penanganan disetujui” <i>Use case</i> selesai <p>Variant 5b: Tidak</p>

Tabel 4.13 Spesifikasi *use case* mengkonfirmasi laporan penanganan

	<ol style="list-style-type: none"> 1. Aktor memilih konfirmasi “Tidak” 2. Sistem menampilkan <i>form</i> pesan 3. Aktor mengisi <i>form</i> alasan mengapa laporan penanganan tidak diterima 4. Sistem mengirim pesan alasan laporan penanganan tidak diterima ke petugas lapangan 5. <i>Use case</i> selesai
Post-Conditions	Status pengaduan berubah menjadi “pengaduan sudah dilakukan penanganan” dan sistem menampilkan notifikasi “laporan penanganan sudah dikonfirmasi”.

4.4.8 Spesifikasi *Use case* Menanggapi Pengaduan

Tabel 4.14 Spesifikasi *use case* menanggapi pengaduan

Use case name	Menanggapi pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor menanggapi pengaduan
Actor	Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu pengaduan. 2. Sistem menampilkan daftar pengaduan. 3. Aktor memilih menu tanggap pengaduan. 4. Sistem menampilkan <i>form</i> pengaduan. 5. Aktor mengisi <i>form</i> pengaduan 6. Aktor memilih untuk mengirim pengaduan. 7. Sistem merubah status pengaduan menjadi “pengaduan sudah di tanggap” 8. <i>Use case</i> selesai.
Alternative Flows	<p>Exception 6: kesalahan pengisian data <i>form</i> tidak lengkap</p> <ol style="list-style-type: none"> 1. Sistem akan menampilkan pesan <i>error</i> pada <i>form</i> yang belum dilengkapi atau terdapat kesalahan pengisian. 2. Kembali ke langkah 5
Post-Conditions	Status pengaduan berubah menjadi “pengaduan sudah ditanggapi oleh OPD” dan sistem

Tabel 4.14 Spesifikasi *use case* menanggapi pengaduan

	menampilkan notifikasi “Pengaduan berhasil dikirim”.
--	--

4.4.9 Spesifikasi *Use case* Melihat Pesan

Tabel 4.15 Spesifikasi *use case* melihat pesan

Use case name	Melihat pesan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor melihat notifikasi
Actor	Masyarakat, Pengelola Pengaduan, Petugas Penanganan Pengaduan, Petugas Lapangan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu pesan 2. Sistem menampilkan daftar pesan 3. Aktor memilih operasi mengelola pesan <ul style="list-style-type: none"> • Membuka Pesan : Variant 3a • Menghapus Pesan : Variant 3b
Alternative Flows	<p>Variant 3a: Membuka Pesan</p> <ol style="list-style-type: none"> 1. Aktor memilih pesan yang ingin dibuka 2. Sistem menampilkan detail pesan 3. <i>Use Case</i> selesai <p>Variant 3b: Menghapus pesan</p> <ol style="list-style-type: none"> 1. Aktor mengklik tombol hapus 2. Sistem menampilkan jendela konfirmasi 3. Aktor melakukan konfirmasi hapus pesan. 4. <i>Use case</i> selesai.
Post-Conditions	Sistem menampilkan daftar pesan

4.4.10 Spesifikasi *Use case* Melihat Status Pengaduan

Tabel 4.16 Spesifikasi *use case* melihat status pengaduan

Use case name	Melihat status pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor melihat status pengaduan
Actor	Masyarakat

Tabel 4.16 Spesifikasi *use case* melihat status pengaduan

Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu pengaduan 2. Sistem menampilkan daftar pengaduan yang telah terkirim
Alternative Flows	-
Post-Conditions	Sistem menampilkan status pengaduan

4.4.11 Spesifikasi *Use case* Mencari Laporan Pengaduan

Tabel 4.17 Spesifikasi *use case* mencari laporan pengaduan

Use case name	Menyampaikan tanggapan pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor menyampaikan tanggapan pengaduan
Actor	Pengunjung
Pre-Conditions	Aktor berada pada halaman awal.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memasukkan kata kunci pada <i>form</i> pencarian. 2. Sistem menampilkan hasil pencarian. 3. <i>Use case</i> selesai.
Alternative Flows	-
Post-Conditions	Sistem menampilkan hasil pencarian.

4.4.12 Spesifikasi *Use case* Melihat *Dashboard* Pengaduan

Tabel 4.18 Spesifikasi *use case* melihat *dashboard* pengaduan

Use case name	Melihat <i>dashboard</i> pengaduan
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor melihat <i>dashboard</i> pengaduan
Actor	Kepala OPD
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor membuka halaman awal <i>admin</i>. 2. Sistem menampilkan pengaduan setiap bulan dalam bentuk diagram. 3. <i>Use case</i> selesai.
Alternative Flows	-
Post-Conditions	Sistem menampilkan data terkait laporan pengaduan.

4.4.13 Spesifikasi Use case Membuat Laporan

Tabel 4.19 Spesifikasi use case membuat laporan

Use case name	Membuat laporan
Brief Description	Use case ini mendeskripsikan tentang bagaimana aktor membuat laporan
Actor	Pengelola Pengaduan, Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Use case ini bermula ketika aktor memilih menu laporan. 2. Sistem akan menampilkan laporan pengaduan bulan ini dalam bentuk tabel. 3. Aktor memasukkan bulan dan tahun pada form sortir. 4. Aktor menekan tombol sortir. 5. Sistem menampilkan hasil sortir. 6. Aktor menekan tombol export to excel. 7. Sistem menampilkan hasil ekspor dalam bentuk unduhan dengan format excel. 8. Use case selesai.
Alternative Flows	-
Post-Conditions	Sistem menampilkan hasil unduhan sortir pengaduan.

4.4.14 Spesifikasi Use case Melihat Laporan Pengaduan

Tabel 4.20 Spesifikasi use case melihat laporan pengaduan

Use case name	Melihat laporan pengaduan
Brief Description	Use case ini mendeskripsikan tentang bagaimana aktor melihat laporan pengaduan
Actor	Masyarakat, Pengelola Pengaduan, Petugas Penanganan Pengaduan, Petugas Lapangan, Kepala OPD
Pre-Conditions	Aktor berada pada halaman awal.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Use case ini bermula ketika aktor membuka halaman awal 2. Sistem menampilkan daftar laporan pengaduan dari semua <i>user</i> yang telah dibalas OPD 3. Use case selesai.
Alternative Flows	Variant 2: fungsi <i>search</i> pengaduan

Tabel 4.20 Spesifikasi *use case* melihat laporan pengaduan

	<ol style="list-style-type: none"> 1. Aktor memasukkan kata kunci terkait pengaduan yang ingin di cari pada <i>form</i> pencarian. 2. Sistem menampilkan hasil pencarian sesuai kata kunci. 3. Melanjutkan langkah 2
Post-Conditions	Sistem menampilkan laporan pengaduan.

4.4.15 Spesifikasi *Use case* Registrasi

Tabel 4.21 Spesifikasi *use case* registrasi

Use case name	Registrasi
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor melakukan registrasi
Actor	Masyarakat
Pre-Conditions	Aktor berada pada halaman awal.
Basic Flows of Events	<ol style="list-style-type: none"> 1. <i>Use case</i> ini bermula ketika aktor memilih menu registrasi 2. Sistem menampilkan <i>form</i> registrasi. 3. Aktor mengisi <i>form</i> data registrasi 4. Aktor memilih untuk mendaftarkan sebagai <i>user</i>. 5. Sistem menyimpan data registrasi. 6. <i>Use case</i> selesai.
Alternative Flows	Exception 4: Kesalahan pengisian data registrasi / tidak lengkap <ol style="list-style-type: none"> 1. Sistem akan menampilkan pesan <i>error</i> pada <i>form</i> yang belum dilengkapi atau terdapat kesalahan pengisian 2. Kembali ke tahap 3
Post-Conditions	Data registrasi akan disimpan ke dalam sistem.

4.4.16 Spesifikasi *Use case* Mengelola Akun

Tabel 4.22 Spesifikasi *use case* mengelola akun

Use case name	Mengelola akun
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor mengelola akun

Tabel 4.22 Spesifikasi use case mengelola akun

Actor	Pengelola Pengaduan, Petugas Penanganan Pengaduan
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Aktor memilih menu daftar akun 2. Sistem menampilkan daftar akun 3. Aktor memilih operasi mengelola akun: <ul style="list-style-type: none"> • Menghapus akun: Variant 3a • Mengedit akun: Variant 3b • Menambah akun: Variant 3c
Alternative Flows	<p>Variant 3a: Menghapus akun</p> <ol style="list-style-type: none"> 1. Aktor memilih menu menghapus akun. 2. Sistem menampilkan jendela konfirmasi. 3. Aktor melakukan konfirmasi hapus akun. 4. Sistem menghapus akun terkait. 5. <i>Use case</i> selesai. <p>Variant 3b: Mengedit akun</p> <ol style="list-style-type: none"> 1. Aktor memilih menu mengedit akun 2. Sistem menampilkan <i>form</i> edit akun. 3. Aktor mengedit data akun. 4. Aktor mengkonfirmasi edit akun. 5. <i>Use case</i> selesai. <p>Variant 3c: Menambah akun</p> <ol style="list-style-type: none"> 1. Aktor memilih menu menambah akun 2. Sistem menampilkan <i>form</i> tambah akun. 3. Aktor mengisi <i>form</i>. 4. Aktor mengkonfirmasi tambah akun. 5. <i>Use case</i> selesai.
Post-Conditions	Sistem menampilkan notifikasi akun berhasil diubah

4.4.17 Spesifikasi Use case Login

Tabel 4.23 Spesifikasi use case login

Use case name	Login
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor mendapatkan otorisasi
Actor	Masyarakat, Administrator dan OPD
Pre-Conditions	Aktor berada pada tampilan awal.
Basic Flows of Events	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>login</i> 2. Sistem menampilkan <i>form login</i>. 3. Aktor mengisi <i>username</i> dan <i>password</i>

	4. Aktor memilih tombol <i>login</i> 5. Sistem mengarahkan aktor ke halaman beranda. 6. <i>Use case</i> selesai.
Alternative Flows	Exception 4: Username / Password Salah 1. Sistem akan menampilkan pesan gagal <i>login</i> 2. Kembali ke tahap 3
Post-Conditions	Aktor masuk ke halaman beranda dan dapat menggunakan fungsi-fungsi dalam sistem sesuai otorisasinya.

4.4.18 Spesifikasi Use case Logout

Tabel 4.24 Spesifikasi *use case logout*

Use case name	Logout
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor keluar dari sistem dan menghapus otorisasinya
Actor	Masyarakat, Administrator dan OPD
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	1. Aktor memilih menu <i>logout</i> 2. Sistem mengarahkan aktor ke halaman awal dan menghapus otorisasinya. 3. <i>Use case</i> selesai.
Alternative Flows	-
Post-Conditions	Aktor keluar dari sistem dan diarahkan ke halaman awal.

4.4.19 Spesifikasi Use case Memperbarui Akun

Tabel 4.25 Spesifikasi *use case memperbarui akun*

Use case name	Memperbarui akun
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor memperbarui biodata
Actor	Masyarakat
Pre-Conditions	Aktor sudah <i>login</i> dan divalidasi oleh sistem.
Basic Flows of Events	1. Aktor memilih menu <i>setting</i> 2. Sistem menampilkan <i>form</i> edit akun 3. Mengedit data sesuai <i>form</i> 4. Aktor memilih simpan

	5. Sistem menyimpan perubahan 6. <i>Use case</i> selesai.
Alternative Flows	Exception 4: kesalahan pengisian data / tidak lengkap 1. Sistem akan menampilkan pesan error pada form yang belum dilengkapi atau terdapat kesalahan pengisian. 2. Kembali ke tahap 3
Post-Conditions	Data akun dari aktor berubah sesuai perubahan yang dilakukan.



BAB 5 PERANCANGAN

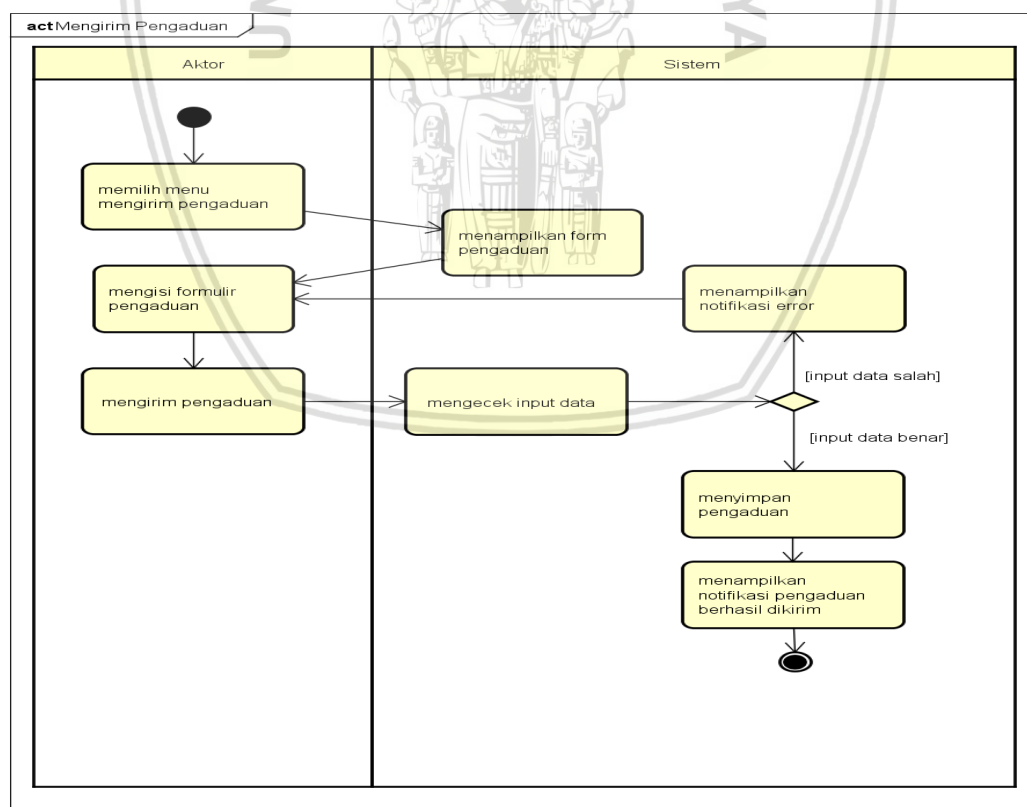
Bab ini menjelaskan tentang gambaran umum sistem dan perancangan dari sistem yang akan diimplementasikan, terdiri dari *activity diagram*, *sequence diagram*, *class diagram*, perancangan basis data, perancangan antarmuka, perancangan komunikasi data, dan perancangan komponen.

5.1 Activity diagram

Activity diagram merupakan diagram yang digunakan untuk menggambarkan alur kerja dari sebuah *use case* berdasarkan spesifikasi *use case* yang telah dijelaskan pada sub bab sebelumnya. *Activity diagram* yang digunakan dalam bab ini hanya beberapa *use case* saja yaitu mengirim pengaduan, verifikasi pengaduan, menyalurkan pengaduan, menangani pengaduan, mendelegasikan petugas lapangan.

5.1.1 Activity diagram Mengirim Pengaduan

Aktivitas yang berjalan pada *use case* mengirim pengaduan dapat dilihat pada Gambar 4.4. Aktivitas tersebut menggambarkan bagaimana proses mengirim pengaduan yang mengikuti proses *basic flow* dan *alternative flow* pada spesifikasi *use case*.

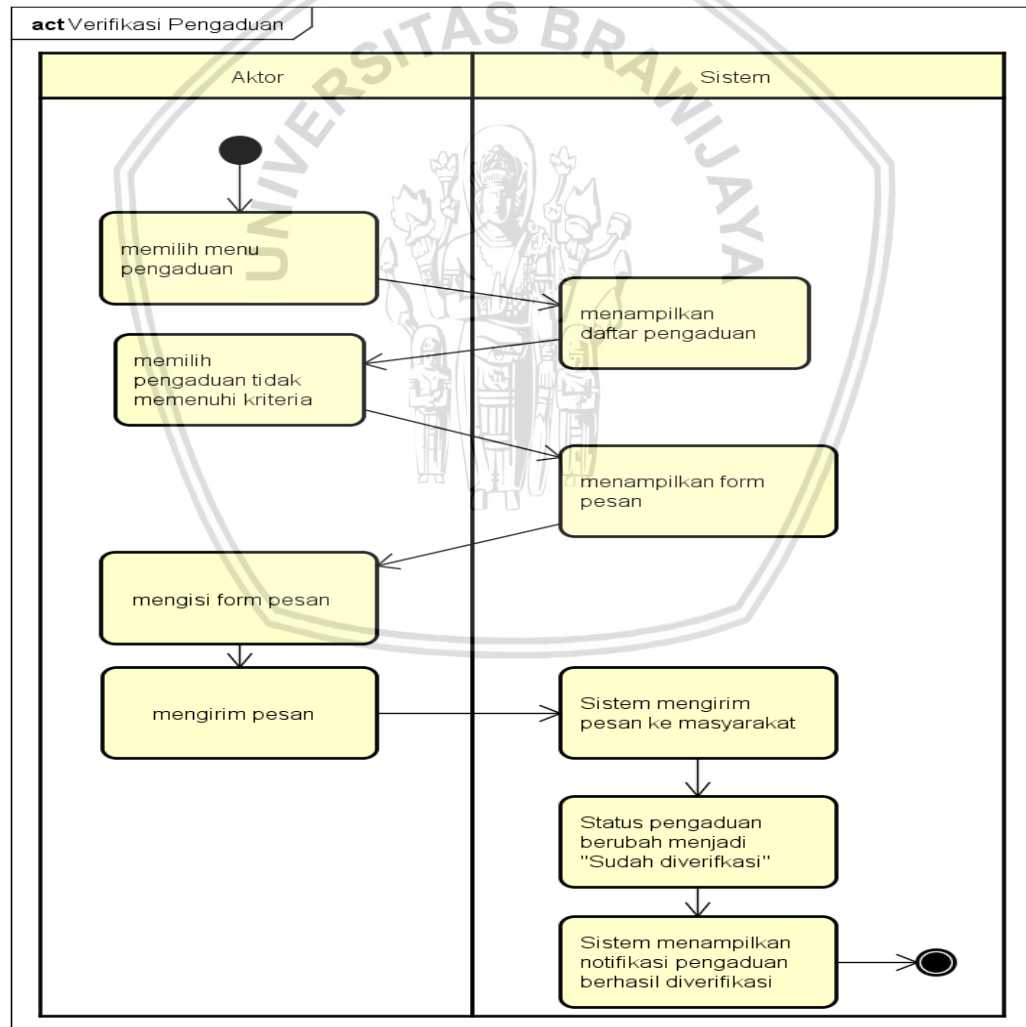


Gambar 5.1 Activity diagram mengirim pengaduan

Pada *activity diagram* mengirim pengaduan, aktor harus memilih menu mengirim pengaduan pada daftar menu yang tersedia. Setelah itu sistem akan menampilkan *form* mengirim pengaduan. Setelah itu aktor harus memasukkan kategori pengaduan, judul pengaduan dan isi pengaduan pada formulir pengaduan yang disediakan. Kemudian sistem akan mengecek data yang dimasukkan oleh user. Jika data yang dimasukkan benar sesuai dengan format dan lengkap maka data pengaduan akan disimpan ke *database*. Setelah itu sistem akan menampilkan notifikasi sukses. Jika tidak maka aktor harus mengisi formulir pengaduan lagi.

5.1.2 Activity diagram Verifikasi Pengaduan 3b: Tidak Memenuhi Kriteria

Pada Gambar 4.5 merupakan *activity diagram* dari *use case* verifikasi pengaduan pada variant 3b : tidak memenuhi kriteria. Diagram tersebut menggambarkan alur yang dilakukan saat pengelola pengaduan melakukan verifikasi pengaduan sesuai dengan spesifikasi *use case* yang telah dibuat.

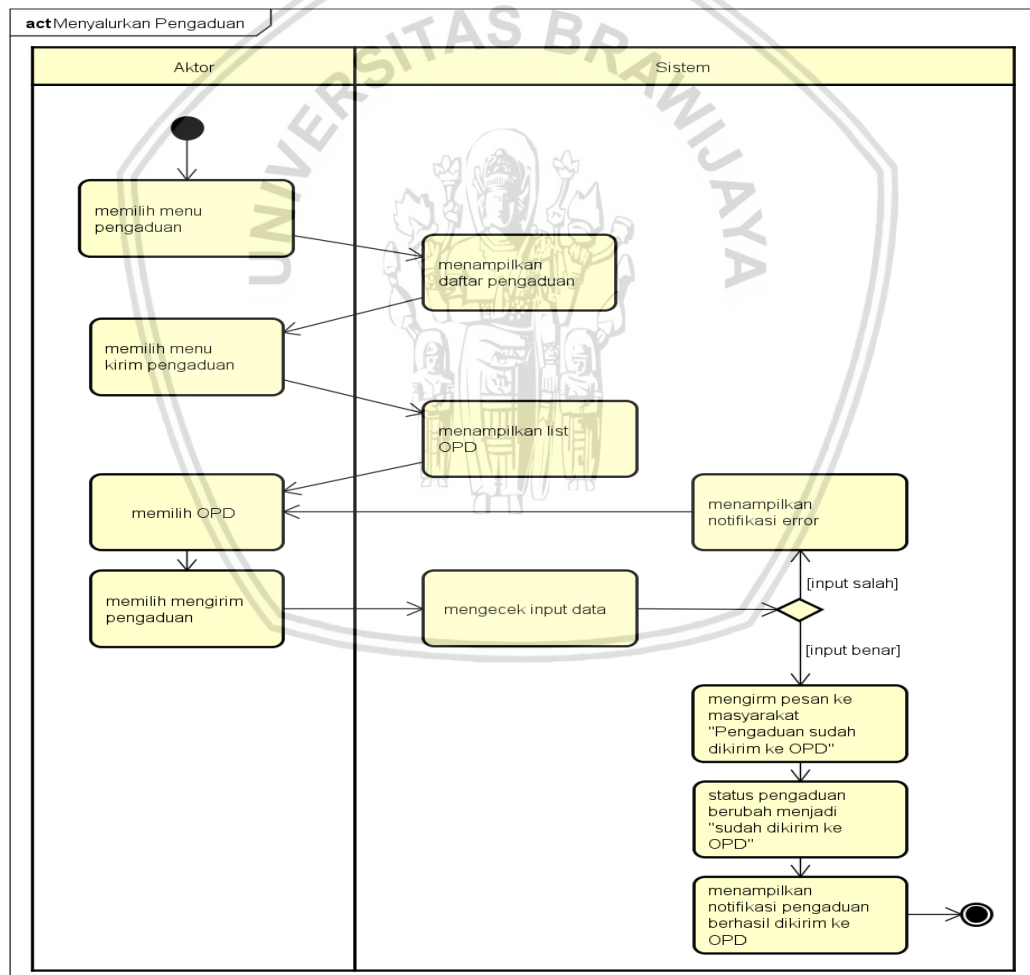


Gambar 5.2 Activity diagram verifikasi pengaduan 3b: tidak memenuhi kriteria

Pada *activity diagram* verifikasi pengaduan 3b: tidak memenuhi kriteria , aktor harus memilih menu pengaduan pada daftar menu yang tersedia. Setelah itu sistem akan menampilkan daftar pengaduan yang belum diverifikasi. Setelah itu aktor menekan tombol tidak memenuhi kriteria. Setelah itu sistem akan menampilkan *form* pesan. Kemudian aktor mengisi alasan mengapa pengaduan tidak memenuhi kriteria pada *form* yang tersedia dan mengirimkannya. Setelah itu sistem akan mengirimkan pesan tersebut ke masyarakat dan merubah status pengaduan menjadi “Sudah diverifikasi”. Setelah berhasil diverifikasi, sistem akan menampilkan notifikasi pengaduan berhasil diverifikasi.

5.1.3 Activity diagram Menyalurkan Pengaduan

Pada Gambar 4.6 merupakan *activity diagram* dari *use case* menyalurkan pengaduan. Diagram tersebut menggambarkan alur yang dilakukan saat pengelola pengaduan mengirim pengaduan ke OPD sesuai dengan spesifikasi *use case* yang telah dibuat.

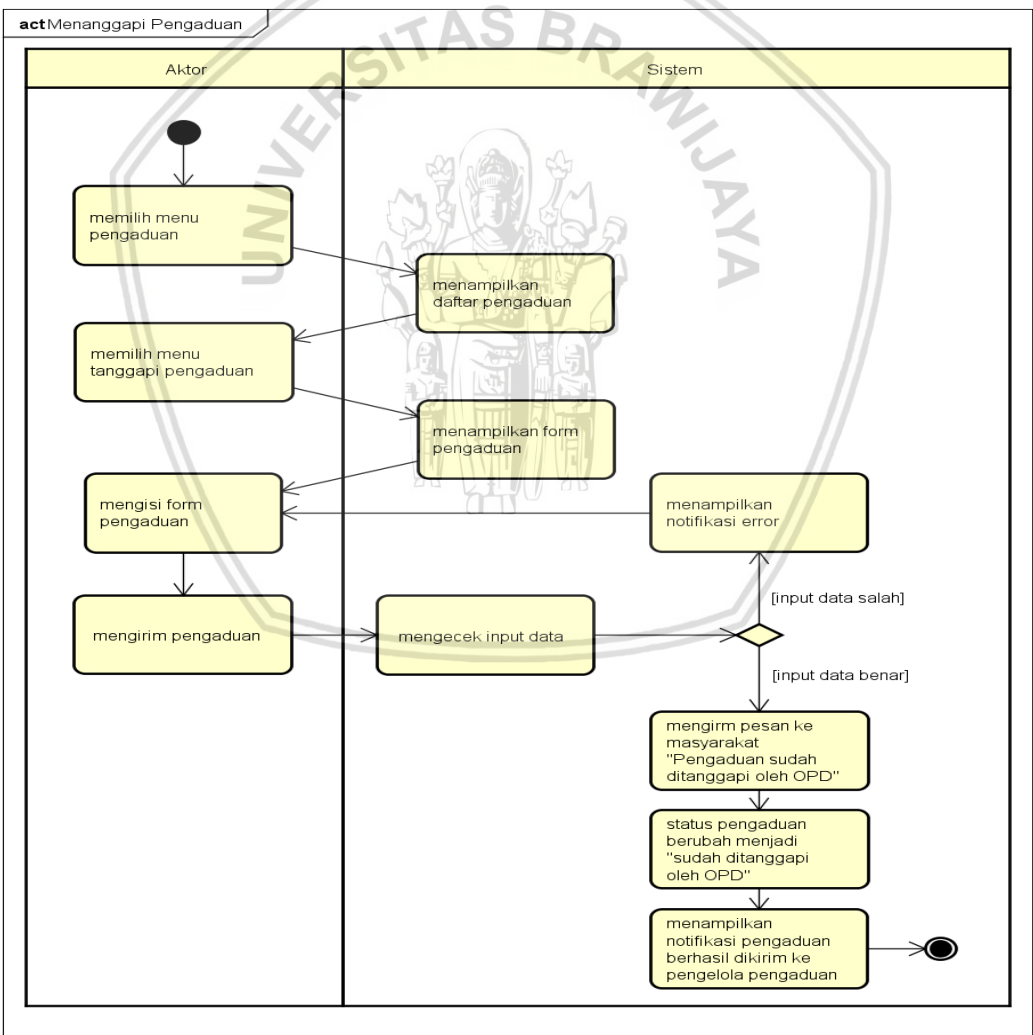


Gambar 5.3 Activity diagram menyalurkan pengaduan

Pada *activity diagram* menyalurkan pengaduan, aktor harus memilih menu pengaduan pada daftar menu yang tersedia. Setelah itu sistem akan menampilkan daftar pengaduan yang belum dikirim ke OPD. Setelah itu aktor menekan tombol kirim pengaduan. Setelah itu sistem akan menampilkan *form* mengirim pengaduan yang berisi list OPD di kota XYZ. Kemudian aktor memilih OPD yang ingin dikirimkan laporan pengaduan untuk ditindaklanjuti. Setelah itu sistem akan mengirimkan pesan ke masyarakat dan merubah status pengaduan menjadi “Sudah dikirim ke OPD”. Setelah berhasil diverifikasi, sistem akan menampilkan notifikasi pengaduan berhasil diverifikasi.

5.1.4 Activity diagram Menanggapi Pengaduan

Pada Gambar 4.7 merupakan *activity diagram* dari *use case* menanggapi pengaduan. Diagram tersebut menggambarkan alur yang dilakukan saat petugas penanganan pengaduan melakukan menanggapi pengaduan sesuai dengan spesifikasi *use case* yang telah dibuat.

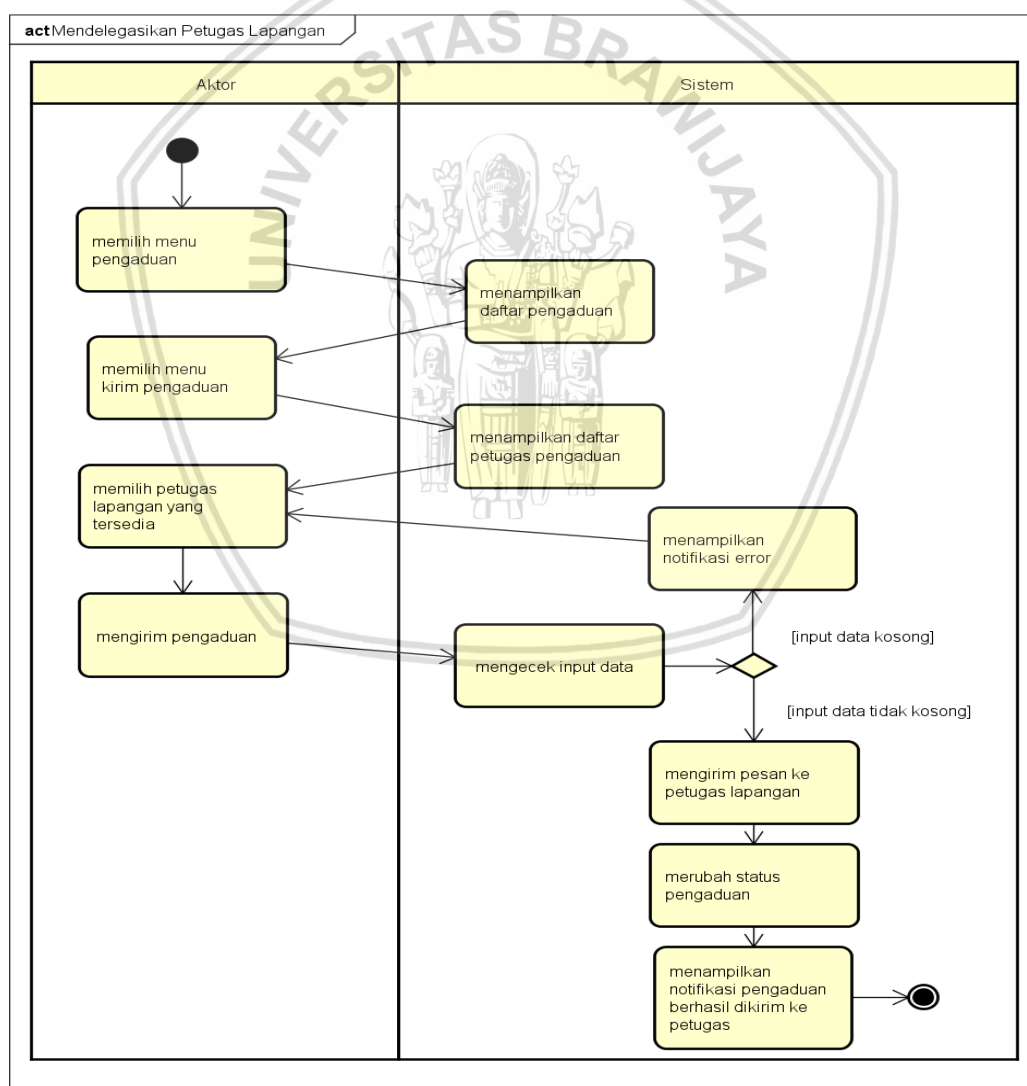


Gambar 5.4 Activity diagram menanggapi pengaduan

Pada *activity diagram* menanggapi pengaduan, aktor harus memilih menu pengaduan pada daftar menu yang tersedia. Setelah itu sistem akan menampilkan daftar pengaduan yang sudah diidentifikasi tidak perlu penanganan lapangan. Setelah itu aktor menekan tombol tanggap pengaduan. Setelah itu sistem akan menampilkan *form* menanggapi. Kemudian aktor memasukkan tanggapan pengaduan pada *form* yang telah disediakan. Setelah itu sistem akan mengirimkan pesan ke masyarakat “pengaduan sudah ditanggapi oleh OPD” dan merubah status pengaduan menjadi “Sudah ditanggapi oleh OPD”. Setelah tanggapan berhasil dikirim, sistem akan menampilkan notifikasi tanggapan berhasil dikirim.

5.1.5 Activity diagram Mendelegasikan Petugas Lapangan

Pada Gambar 4.8 merupakan *activity diagram* dari *use case* mendelegasikan petugas lapangan. Diagram tersebut menggambarkan alur yang dilakukan saat petugas pengelola pengaduan menugaskan petugas lapangan untuk melakukan penanganan di lapangan dengan cara mengirim laporan pengaduan.



Gambar 5.5 Activity diagram mendelegasikan petugas lapangan

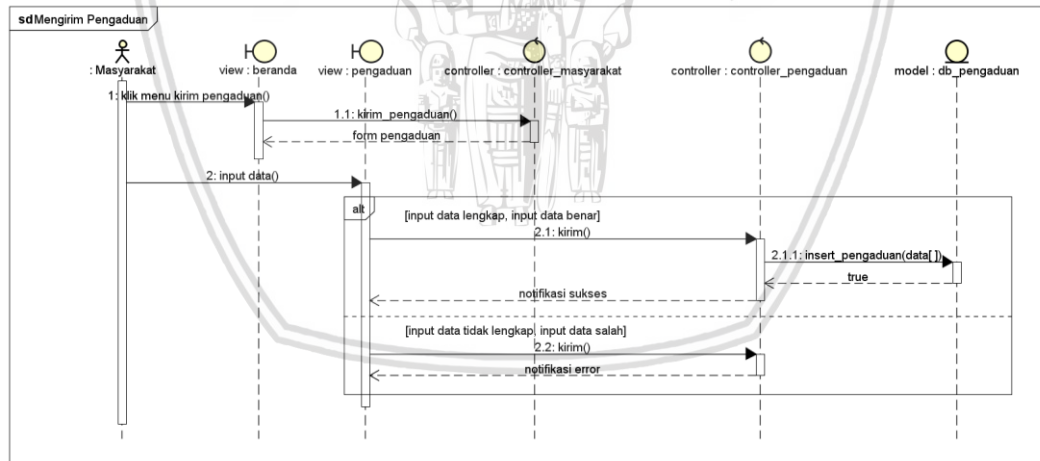
Pada activity diagram mendelegasikan petugas lapangan, aktor harus memilih menu pengaduan pada daftar menu yang tersedia. Setelah itu sistem akan menampilkan daftar pengaduan yang sudah diidentifikasi perlu penanganan lapangan. Setelah itu aktor menekan tombol kirim pengaduan. Setelah itu sistem akan menampilkan *form* yang berisi list petugas pengaduan yang tersedia di OPD. Kemudian aktor memilih petugas yang tersedia dan mengirimkan laporan pengaduan. Setelah itu sistem akan mengirimkan pesan ke petugas lapangan dan merubah status pengaduan menjadi “Sedang dilakukan penanganan lapangan oleh OPD”. Setelah laporan pengaduan berhasil dikirim, sistem akan menampilkan notifikasi pengaduan berhasil dikirim.

5.2 Sequence diagram

Di bagian *sequence diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim antara *object* dan juga interaksi antara *object*. *Sequence diagram* yang digunakan dalam bab ini hanya dari beberapa *activity diagram* saja yaitu mengirim pengaduan, menyalurkan pengaduan, menangani pengaduan, mendelegasikan petugas lapangan.

5.2.1 Sequence diagram Mengirim Pengaduan

Sequence diagram ini dibuat menggunakan *activity diagram* dari Gambar 4.4 pada bab sebelumnya sebagai acuan. *Sequence diagram* mengirim pengaduan dapat dilihat pada Gambar 5.1.

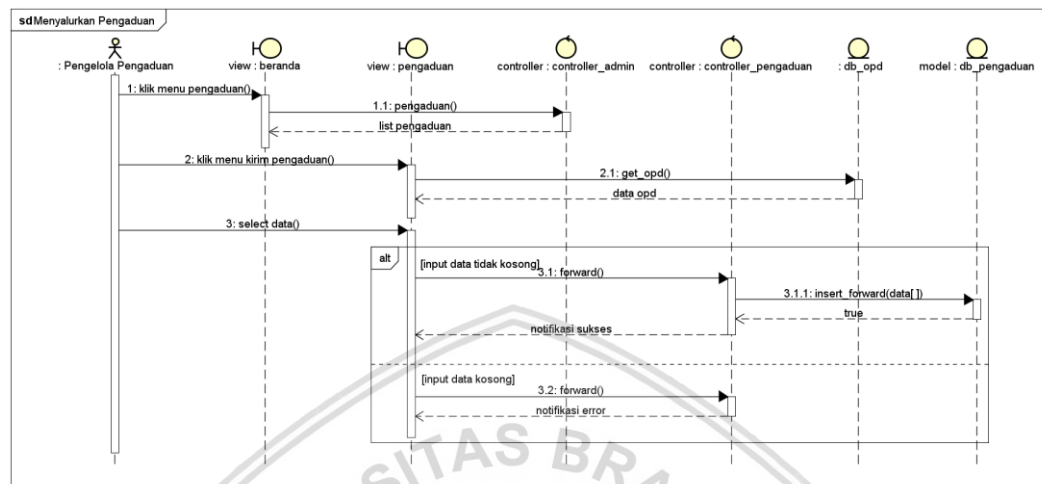


Gambar 5.6 Sequence diagram mengirim pengaduan

Sequence diagram dimulai ketika masyarakat memilih menu mengirim pengaduan pada *boundary* beranda. Beranda memanggil *controller_masyarakat* untuk membuka tampilan *form* mengirim pengaduan. Kemudian masyarakat mengisi *form* dan mengirim data. Apabila *form* tidak diisi secara lengkap atau input tidak benar, maka *controller_pengaduan* akan memberikan notifikasi error. Jika input lengkap dan benar maka *controller_pengaduan* memasukkan data ke database melalui *model db_pengaduan*. Setelah itu *controller_pengaduan* akan memberikan notifikasi sukses.

5.2.2 Sequence diagram Menyalurkan Pengaduan

Sequence diagram ini dibuat menggunakan *activity diagram* dari Gambar 4.6 pada bab sebelumnya sebagai acuan. *Sequence diagram* mengelola pengaduan dapat dilihat pada Gambar 5.2.



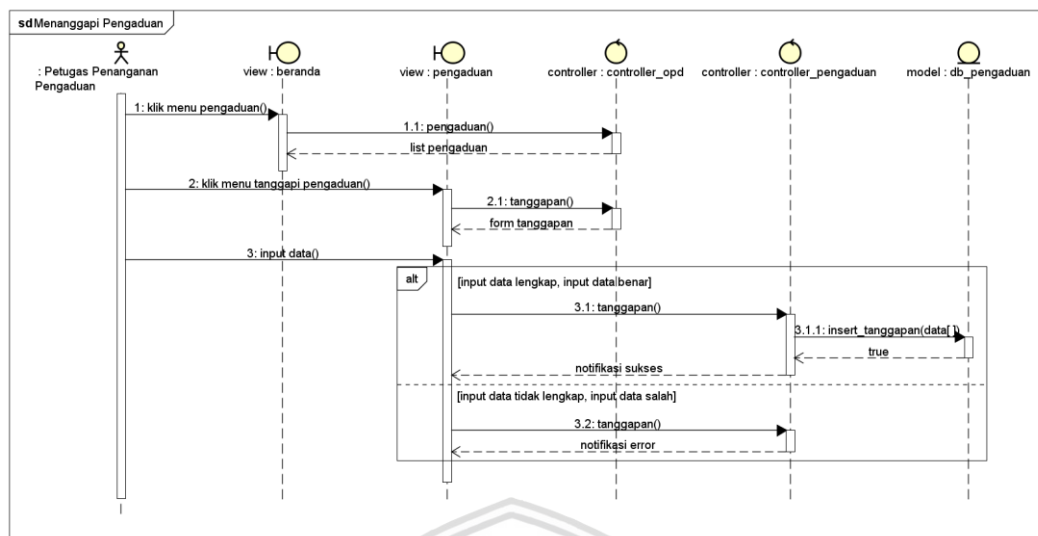
Gambar 5.7 Sequence diagram menyalurkan pengaduan

Sequence diagram dimulai ketika pengelola pengaduan memilih menu pengaduan pada *boundary* beranda. Beranda memanggil *controller_admin* untuk membuka tampilan list pengaduan. Kemudian pengelola pengaduan memilih menu mengirim pengaduan. Tampilan pengaduan akan meminta data list OPD ke *database* melalui model *db_opd* dan menampilkan data list OPD.

Setelah itu pengelola pengaduan memilih list opd yang ingin dikirimkan pengaduan dan mengirim data. Apabila pengelola pengaduan belum memilih OPD, maka *controller_pengaduan* akan memberikan notifikasi *error*. Jika input data tidak kosong *controller_pengaduan* akan memasukkan data ke *database* melalui model *db_pengaduan*. Setelah itu *controller_pengaduan* akan memberikan notifikasi sukses.

5.2.3 Sequence diagram Menanggapi Pengaduan

Sequence diagram ini dibuat menggunakan *activity diagram* dari Gambar 4.7 pada bab sebelumnya sebagai acuan. *Sequence diagram* membalas pengaduan dapat dilihat pada Gambar 5.3.



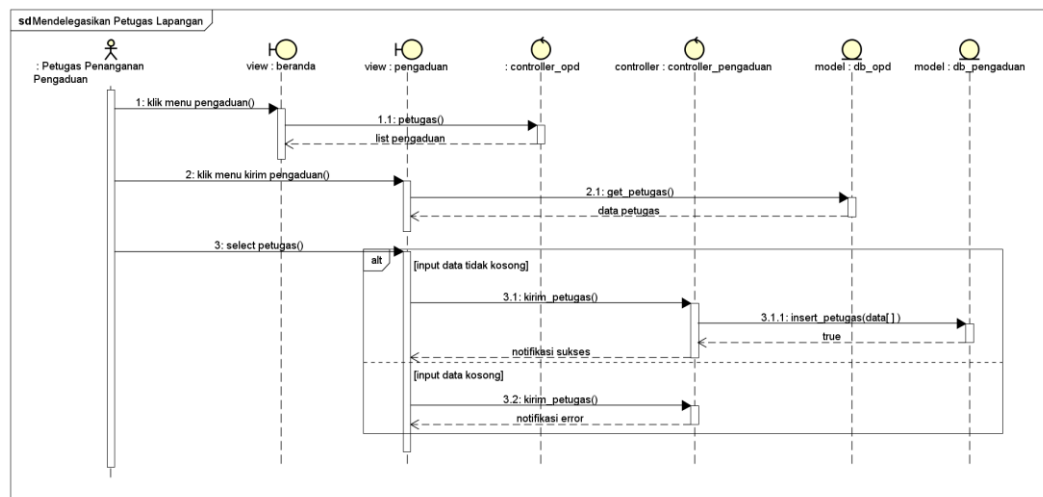
Gambar 5.8 Sequence diagram menanggapi pengaduan

Sequence diagram dimulai ketika petugas penanganan pengaduan memilih menu pengaduan pada *boundary* beranda. Beranda memanggil *controller_opd* untuk menampilkan *list* pengaduan. Kemudian petugas penanganan pengaduan memilih menu tanggap pengaduan dan *controller_opd* akan membuka tampilan *form* menanggapi pengaduan.

Setelah itu petugas penanganan pengaduan mengisi tanggapan pengaduan dan mengirimkan data. Apabila petugas penanganan pengaduan tidak mengisi *form* secara lengkap dan benar, maka *controller_pengaduan* akan memberikan notifikasi error. Jika input data lengkap dan benar maka *controller_pengaduan* akan memasukkan data ke *database* melalui *model db_pengaduan*. Setelah itu *controller_pengaduan* akan memberikan notifikasi sukses.

5.2.4 Sequence diagram Mendelegasikan Petugas Lapangan

Sequence diagram ini dibuat menggunakan *activity diagram* dari Gambar 4.8 pada bab sebelumnya sebagai acuan. *Sequence diagram* mendelegasikan petugas lapangan dapat dilihat pada Gambar 5.4.



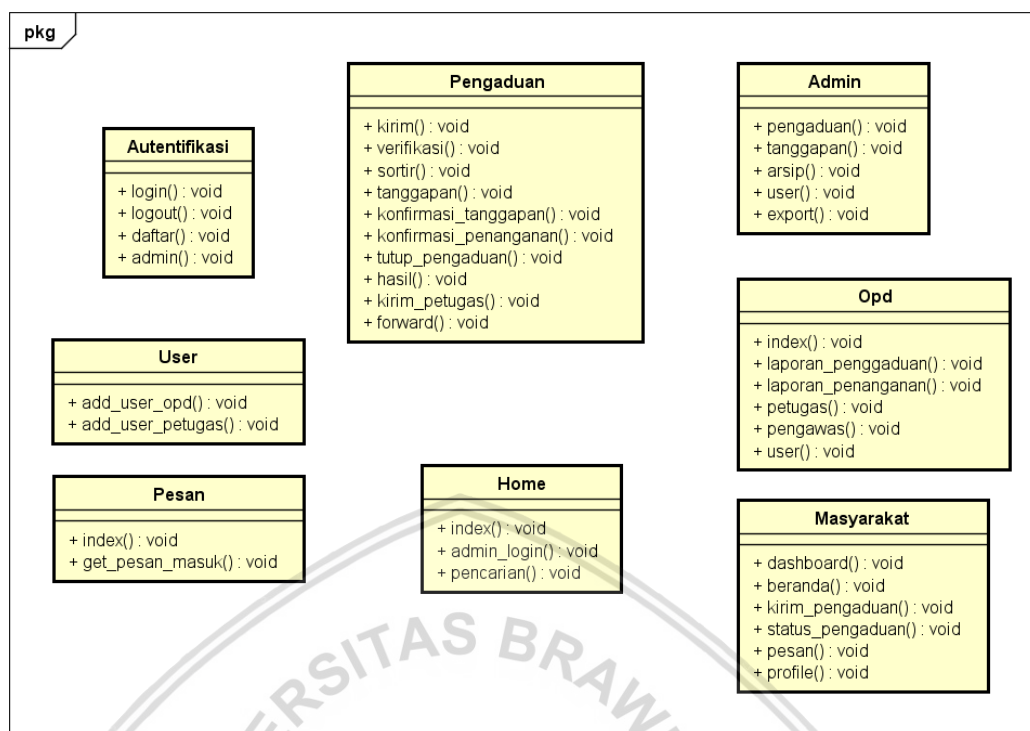
Gambar 5.9 Sequence diagram mendelegasikan petugas lapangan

Sequence diagram dimulai ketika petugas penanganan pengaduan memilih menu pengaduan pada *boundary* beranda. Beranda memanggil *controller_opd* untuk membuka tampilan list pengaduan. Kemudian pengelola pengaduan memilih menu mengirim pengaduan. Tampilan pengaduan akan meminta data list petugas OPD ke *database* melalui model *db_opd* dan menampilkan data list petugas.

Setelah itu pengelola pengaduan memilih petugas yang ingin dikirimkan laporan pengaduan dan mengirim data. Apabila petugas penanganan pengaduan belum memilih petugas OPD, maka *controller_admin* akan memberikan notifikasi *error*. Jika input data tidak kosong *controller_pengaduan* akan memasukkan data ke *database* melalui model *db_pengaduan*. Setelah itu *controller_pengaduan* akan memberikan notifikasi sukses.

5.3 Perancangan Class Diagram

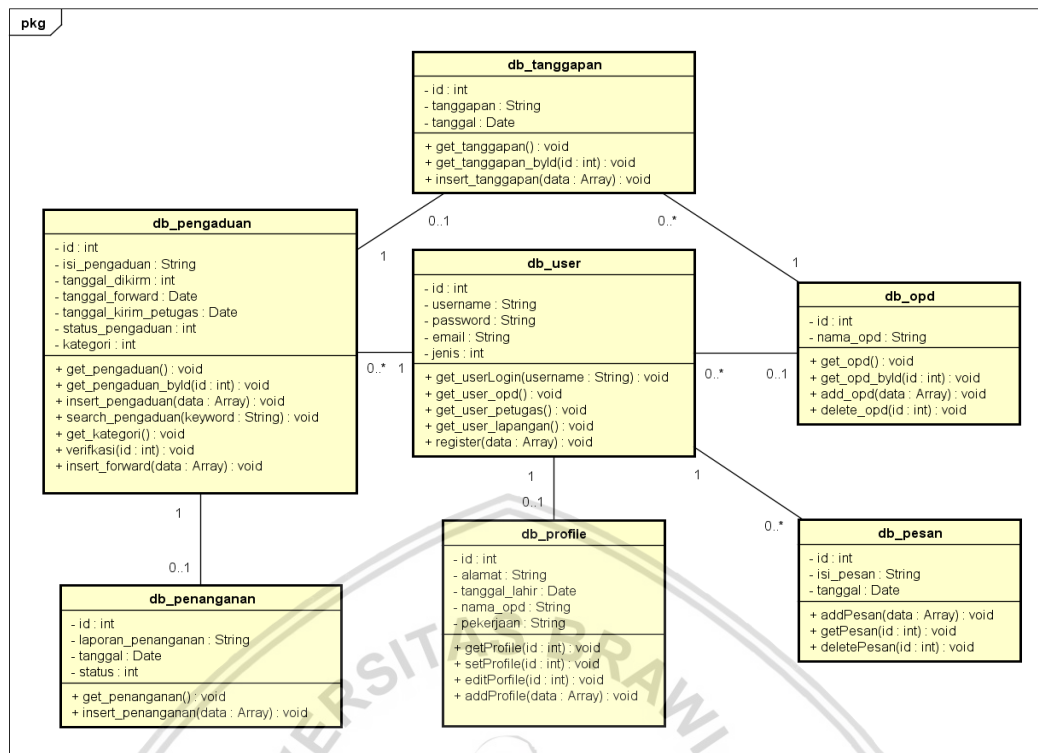
Class diagram menggambarkan struktur dari sistem yang didefinisikan berdasarkan kelas-kelas yang membangun sistem. Perancangan *class diagram* dapat dilihat dalam Gambar 5.5.



Gambar 5.10 Class diagram sebagai domain controller

Pada Gambar 5.5 digambarkan *class diagram* yang digunakan dalam perancangan sistem. Terdapat 8 *class* yakni Authentication yang memiliki peran dalam masalah autentifikasi, Pengaduan yang memiliki peran untuk menangani data-data pengaduan, Admin memiliki peran dalam proses mengelola data dan tampilan pengaduan, OPD memiliki peran dalam proses penanganan pengaduan, User memiliki peran dalam pengelolaan *user*, Pesan memiliki peran dalam pengelolaan pesan dan Masyarakat memiliki peran dalam mengirim pengaduan dan menampilkan status pengaduan.

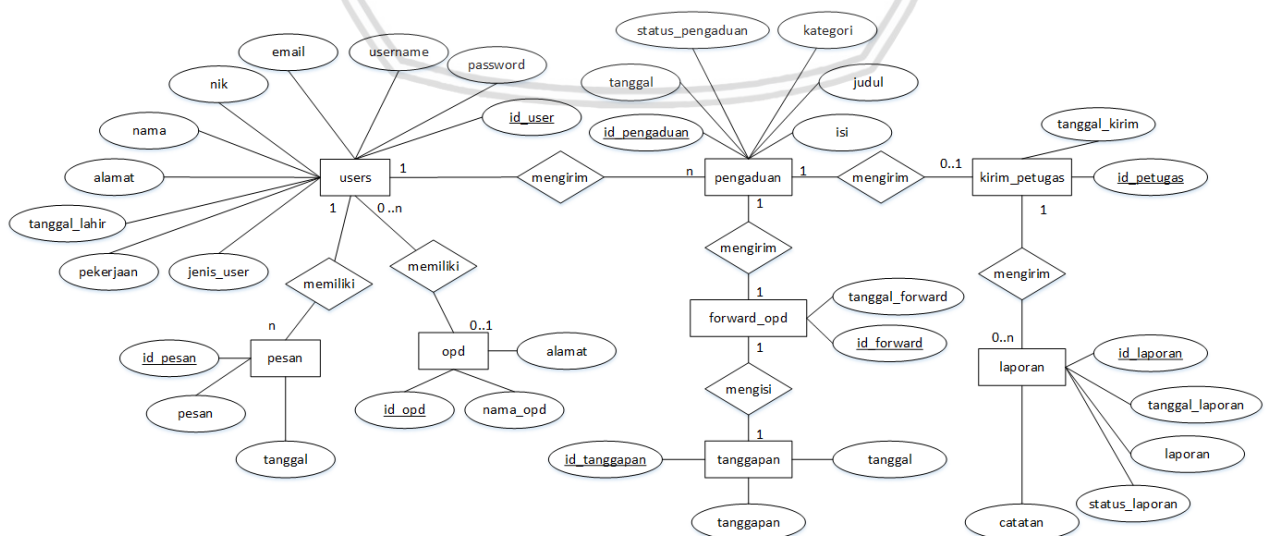
Pada Gambar 5.6 digambarkan *class diagram* yang akan digunakan sebagai pemodelan *database*. Terdapat 7 *class* yakni *db_pengaduan*, *db_user*, *db_tanggapan*, *db_opd*, *db_penanganan*, *db_pesan* dan *db_profile*.



Gambar 5.11 Class diagram sebagai domain model

5.4 Perancangan Basis Data

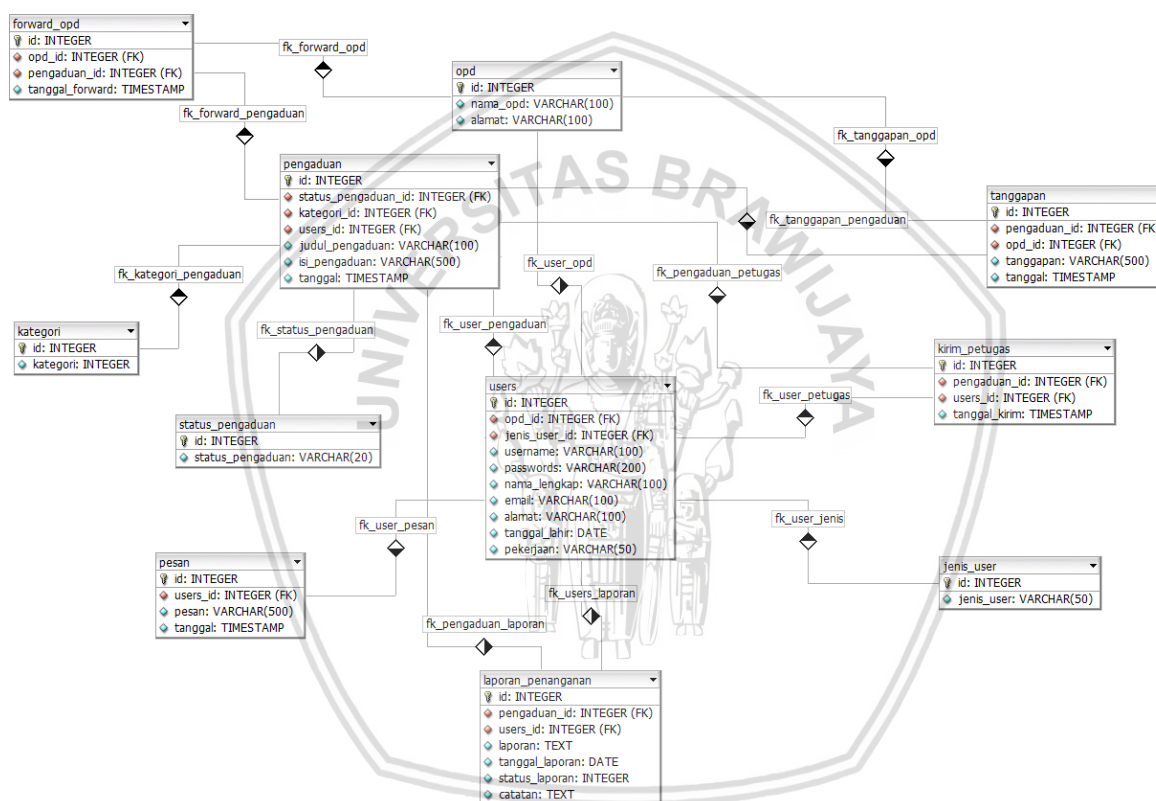
Perancangan basis data dibuat dengan mendefinisikan struktur-struktur pada tabel yang terdapat pada *class diagram* domain model. Perancangan basis data digunakan untuk mempermudah implementasi basis data ke dalam bentuk tabel di *database*. Perancangan basis data secara logika diwujudkan dalam bentuk *Entity Relational Diagram* (ERD) konseptual. ERD konseptual dari sistem dapat dilihat pada Gambar 5.12.



Gambar 5.12 Entity Relational Diagram (ERD)

ERD konseptual dibuat dengan menentukan entitas pada sistem. Terdapat 8 entitas yang dapat dibentuk dari sistem ini, yaitu users, opd, pesan, pengaduan, forward_opd, tanggapan, kirim_petugas, dan laporan. Pada ERD yang dibuat entitas users memiliki relasi dengan opd *one to one*, entitas users dengan pesan *one to many*, users dengan pengaduan *one many*. Entitas pengaduan dengan forward_opd *one to one*, entitas pengaduan dengan kirim_petugas *one to one*. Entitas kirim_petugas dengan laporan *one to many*.

ERD konseptual diimplementasikan secara spesifik ke dalam bentuk *physical data model (PDM)*. PDM dari sistem informasi *E-Complaint* dapat dilihat pada Gambar 5.13.



Gambar 5.13 Physical Data Model

Dari hasil implementasi PDM didapatkan 9 tabel yang terdiri dari users, pengaduan, forward_pengaduan, tanggapan, kategori, pesan, profile, jenis dan opd. Tabel users digunakan untuk menyimpan data *user*. Tabel pengaduan digunakan untuk menyimpan data pengaduan. Tabel forward_opd digunakan untuk menyimpan pengaduan dikirim ke OPD mana. Tabel OPD digunakan untuk menyimpan data OPD. Tabel tanggapan digunakan untuk menyimpan data tanggapan pengaduan.

5.5 Perancangan Antarmuka

Bagian ini menjelaskan bagaimana antarmuka sistem dirancang untuk membantu dalam proses implementasi sistem. Perancangan antarmuka dibagi menjadi beberapa menu yang disesuaikan dengan kebutuhan sistem yang telah dijelaskan pada bab sebelumnya.

5.5.1 Halaman Awal

Perancangan antarmuka halaman awal dapat dilihat pada Gambar 5.8.

Logo Login Register

Search Pengaduan

Laporan Pengaduan Tanggal Pengiriman

Tanggapan pengaduan
Oleh : OPD Tanggal dibalas

Laporan Pengaduan Tanggal Pengiriman

Tanggapan pengaduan
Oleh : OPD Tanggal dibalas

Footer

Gambar 5.14 Perancangan halaman awal

Halaman awal dapat digunakan untuk mencari pengaduan dan dapat menampilkan pengaduan-pengaduan terbaru. Terdapat juga *navbar* untuk *login* dan *registrasi* bagi yang belum memiliki akun.

5.5.2 Halaman Beranda *User*

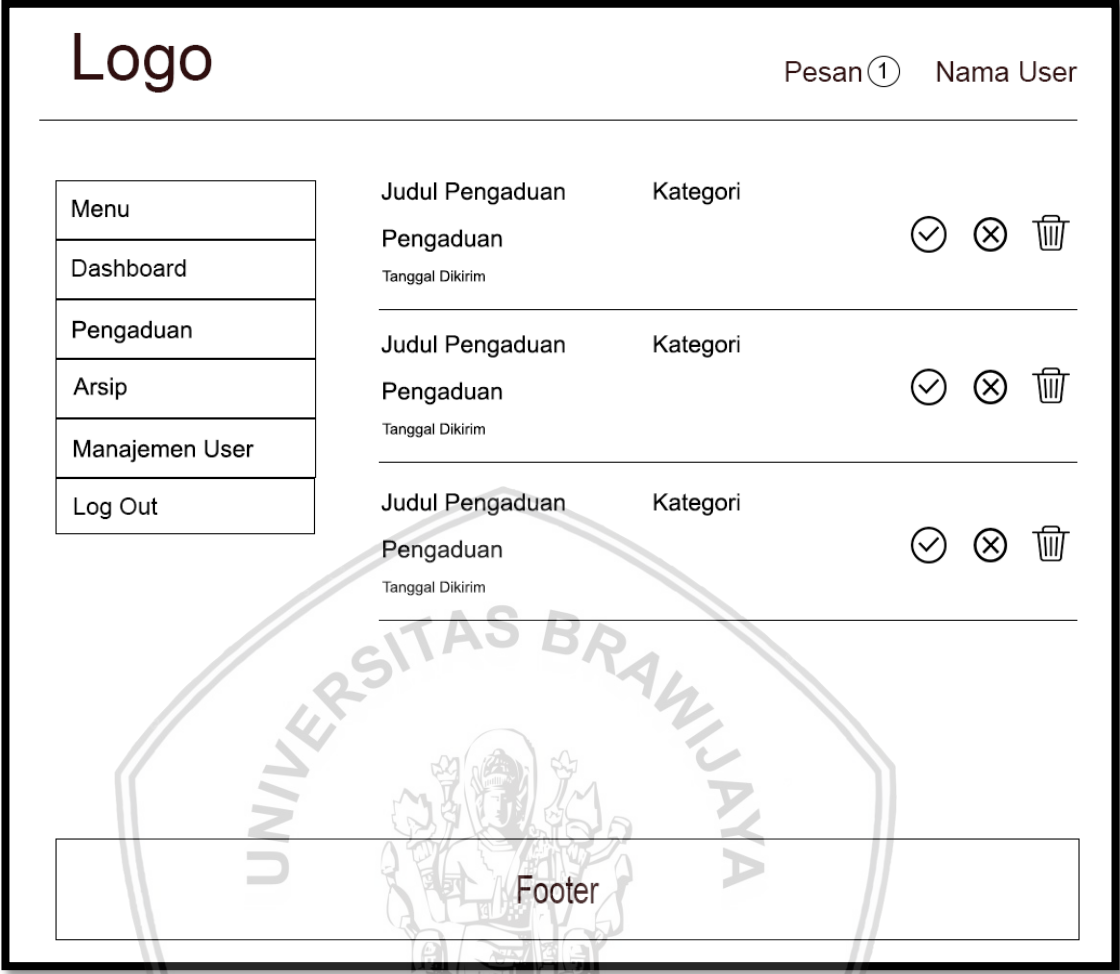
Perancangan antarmuka halaman beranda *user* dapat dilihat pada Gambar 5.9.

Gambar 5.15 Perancangan halaman beranda *user*

Pada perancangan antar muka halaman beranda *user* terdapat *navbar* untuk melihat pesan masuk dan melihat profil *user*. Terdapat menu dibagian kiri yaitu beranda, kirim pengaduan, pengaduan dan *log out*. Halaman ini dapat juga digunakan untuk mencari pengaduan dan dapat menampilkan pengaduan-pengaduan terbaru.

5.5.3 Halaman Pengaduan *Admin*

Perancangan antarmuka halaman pengaduan pada *admin* dapat dilihat pada Gambar 5.10.

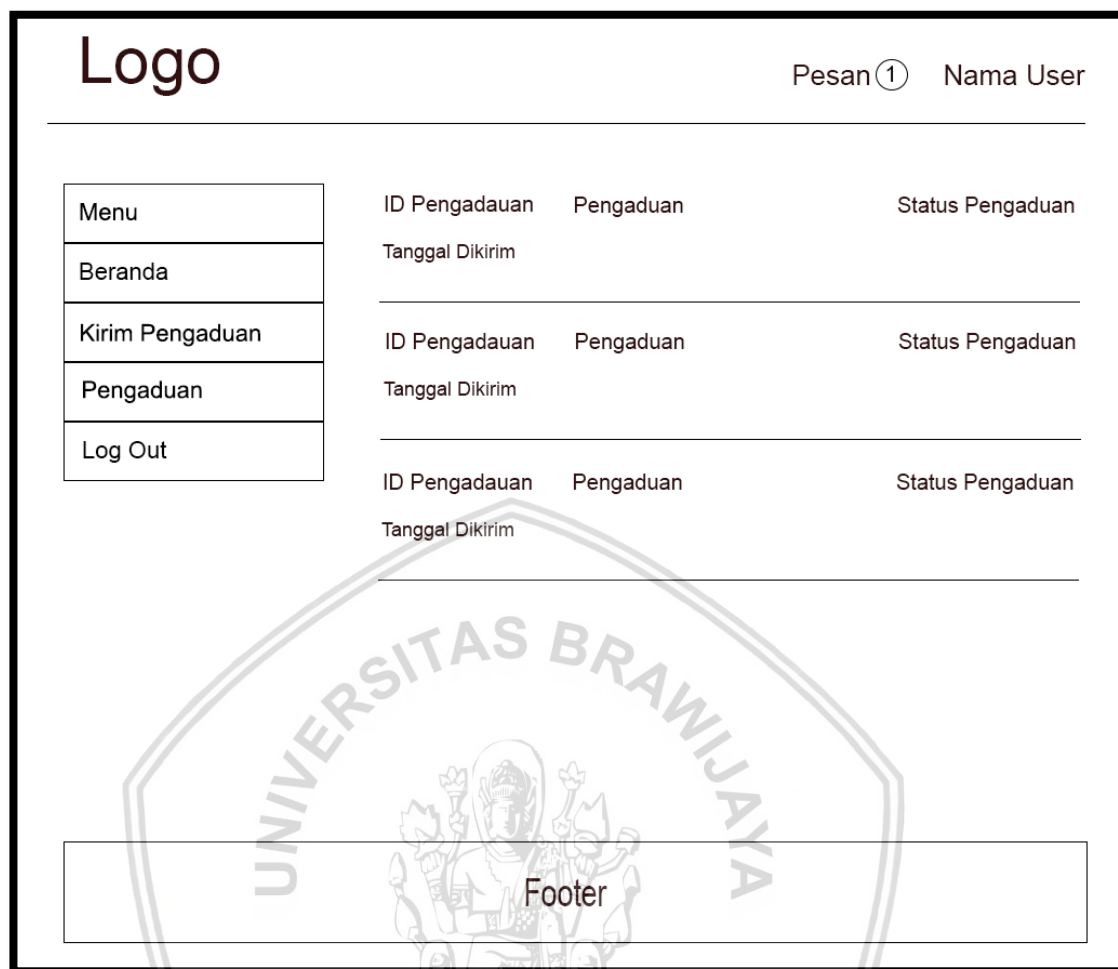


Gambar 5.16 Perancangan halaman pengaduan *admin*

Pada perancangan antar muka halaman pengaduan *admin* terdapat *navbar* untuk melihat pesan masuk dan melihat profil *admin*. Terdapat menu dibagian kiri yaitu dashboard, pengaduan, arsip, manajemen *user* dan *log out*. Halaman ini terdapat pengaduan-pengaduan terbaru dari masyarakat beserta detail pengaduan seperti kategori dan tanggal dikirim. Dibagian sebelah kanan terdapat tombol verifikasi pengaduan dan menghapus pengaduan.

5.5.4 Halaman Pengaduan *User*

Perancangan antarmuka halaman pengaduan pada *user* dapat dilihat pada Gambar 5.11.



Gambar 5.17 Perancangan halaman pengaduan *user*

Pada perancangan antar muka halaman beranda *user* terdapat *navbar* untuk melihat pesan masuk dan melihat profil *user*. Terdapat menu dibagian kiri yaitu beranda, kirim pengaduan, pengaduan dan *log out*. Halaman ini dapat menampilkan pengaduan-pengaduan yang telah dikirim oleh *user* dan melihat status pengaduan mereka.

5.6 Perancangan Algoritme

Pada perancangan algoritme dijelaskan algoritme yang dipakai untuk diimplementasikan. Terdapat beberapa algoritme program yang akan dijelaskan dalam sub bab ini yakni algoritme mengirim pengaduan, verifikasi pengaduan.

5.6.1 Algoritme Mengirim Pengaduan

Dalam proses penambahan artikel terdapat tahap-tahap yang dilakukan oleh sistem. Pada Tabel 5.1 dapat dilihat perancangan algoritme yang diimplementasikan pada sistem.

Tabel 5.1 Perancangan algoritme mengirim pengaduan

No	Pseudocode
1	Mulai
2	Jalankan fungsi mengirim pengaduan
3	Mendapatkan data pengaduan yang sudah diinputkan
4	Jika input salah
5	Kembali ke halaman sebelumnya dengan pesan input yang salah
6	Akhir pengkondisian
7	Lainnya
8	Memanggil fungsi memasukkan data pengaduan ke database
9	Akhir pengkondisian
10	Kembali ke halaman pengaduan dengan pesan berhasil mengirim
11	pengaduan
12	Selesai

5.6.2 Algoritme Verifikasi Pengaduan

Pada Tabel 5.2 dapat dilihat perancangan algoritme yang digunakan untuk verifikasi pengaduan.

Tabel 5.2 Perancangan algoritme verifikasi pengaduan

No	Pseudocode
1	Mulai
2	Jalankan fungsi verifikasi pengaduan
3	Mendapatkan data pengaduan sesuai id yang dipilih
4	Jika input memenuhi kriteria
5	Mengirim notifikasi ke masyarakat pengaduan memenuhi kriteria
6	Akhir pengkondisian
7	Lainnya
9	Mengirim notifikasi ke masyarakat pengaduan tidak memenuhi kriteria
10	Akhir pengkondisian
11	Menampilkan pesan pengaduan berhasil diverifikasi
12	Selesai
13	

BAB 6 IMPLEMENTASI

Bab ini membahas tentang tahap implementasi aplikasi berdasarkan analisis kebutuhan dan perancangan yang telah dibahas pada bab sebelumnya. Bab ini dibagi menjadi beberapa pembahasan, antara lain spesifikasi sistem, Batasan-batasan implementasi, implementasi basis data, implementasi kelas, dan implementasi antarmuka.

6.1 Spesifikasi Sistem

Pada sub bab ini menjelaskan mengenai spesifikasi minimum *web server* yang digunakan agar sistem ini berjalan secara optimal. Spesifikasi minimum tersebut dapat dilihat pada Tabel 6.1

Tabel 6.1 Tabel spesifikasi minimum *web server*

Web Server	Apache versi 2.4.29
DBMS	Mysql Ver 5.0.12.
PHP	versi 7.0.8

6.2 Impementasi *Class Diagram*

Pada bagian ini dilakukan implementasi dari perancangan *class diagram* yang dibuat pada bab sebelumnya. Implementasi *class diagram* dilakukan dengan cara merubah kelas-kelas pada perancangan *class diagram* menjadi *file-file* dengan tipe php dan menuliskan atribut beserta metodenya kedalam bentuk *source code*. Class diagram pada domain controller diimplementasikan menjadi Authentifikasi.php, Pengaduan.php, Admin.php, User.php, Pesan.php, Home.php, Opd.php, Masyarakat.php. Sedangkan tabel pada *class diagram* untuk domain model diimplementasikan menjadi db_tanggapan.php, db_pengaduan, db_user, db_opd, db_penanganan, db_profile, db_pesan.

6.2.1 Implementasi *class diagram* kelas pengaduan

Pada perancangan *class diagram* domain controller terdapat kelas pengaduan yang dapat dilihat pada Gambar 5.10. Kelas pengaduan berfungsi untuk menampilkan tampilan halaman pengaduan, mengolah *input dan output* data pengaduan. Pada kelas pengaduan terdapat metode abstrak kirim, verifikasi, sortir, tanggpan, konfirmasi_tanggapan, tutup_pengaduan, kirim_petugas, dan forward. Kelas dan metode tersebut diimplementasikan ke dalam bentuk *souce code*. Hasil implementasi kelas pengaduan dapat dilihat pada Tabel 6.2.

Tabel 6.2 Implementasi *class diagram* tabel pengaduan

Pengaduan.php	
No. Baris	Source Code
1	<?php
2	defined('BASEPATH') OR exit('No direct script access
3	allowed');

Pengaduan.php	
No. Baris	Source Code
4	
5	class Pengaduan extends CI_Controller {
6	
7	public function kirim(){
8	
9	}
10	
11	public function verifikasi(){
12	}
13	
14	public function sortir(){
15	}
16	
17	public function tanggapan(){
18	}
19	
20	public function konfirmasi_tanggapan(){
21	}
22	
23	public function konfirmasi_penanganan(){
24	}
25	
26	public function tutup_pengaduan(){
27	}
28	
29	public function hasil(){
30	}
31	
32	public function forward(){
33	}
34	
35	}
36	
37	/* End of file pengaduan.php */
38	/* Location: ./application/controllers/pengaduan.php */

6.2.2 Implementasi *class diagram* kelas db_pengaduan

Pada perancangan *class diagram* domain model terdapat kelas db_pengaduan yang dapat dilihat pada Gambar 5.11 . Kelas db_pengaduan berfungsi untuk memproses create, retrieve , update dan delete terkait data pengaduan. Pada kelas pengaduan terdapat atribut id_pengaduan, tanggal_dikirim dan tanggal_forward. Pada kelas pengaduan juga terdapat metode get_pengaduan, insert_pengaduan, search_pengaduan, verifikasi_pengaduan, dan insert_forward. Kelas, atribut, dan metode tersebut diimplementasikan ke dalam bentuk *source code*. Hasil implementasi dari kelas db_pengaduan dapat dilihat pada Tabel 6.3 .

Tabel 6.3 Implementasi *class diagram* tabel db_pengaduan

db_pengaduan.php	
No. Baris	Source Code
1	<?php
2	defined('BASEPATH') OR exit('No direct script access
3	allowed');

db_pengaduan.php	
No. Baris	Source Code
4	
5	var \$id;
6	var \$id_pengaduan;
7	var \$tanggal_dikirim;
8	
9	var \$tanggal_forward;
10	var \$tanggal_kirim_petugas;
11	var \$status_pengaduan;
12	var \$kategori;
13	
14	class Db_pengaduan extends CI_Model {
15	
16	public function get_pengaduan(){
17	
18	
19	\$this->db->from('pengaduan as p');
20	\$this->db->join('kategori as k', 'p.kategori_id = k.id');
21	\$this->db->join('user as u', 'p.user_id = u.id');
22	\$this->db->join('profile as f', 'f.user_id = u.id');
23	\$this->db->order_by('p.id_pengaduan', 'DESC');
24	\$this->db->limit(5);
25	
26	\$query = \$this->db->get();
27	
28	return \$query;
29	
30	\$query->free_result();
31	}
32	
33	
34	public function get_pengaduan_IDbyTanggapan(\$id){
35	\$this->db->where('id_penanganan', \$id);
36	\$query = \$this->db->get('penanganan');
37	
38	return \$query;
39	\$query->free_result();
40	
41	}
42	
43	
44	public function search_pengaduan(\$keyword){
45	
46	}
47	
48	
49	public function get_kategori(){
50	
51	\$query = \$this->db->get('kategori');
52	
53	return \$query;
54	
55	\$query->free_result();
56	
57	}
58	
59	public function insert_pengaduan(\$data){
60	
61	if(\$this->db->insert('pengaduan', \$data)){
62	echo "sukses";
63	}
64	}

db_pengaduan.php	
No. Baris	Source Code
65	
66	public function insert_tanggapan(\$data){
67	
68	if(\$this->db->insert('tanggapan', \$data)){
69	//echo "sukses";
70	}
71	}
72	public function verifikasi(\$id){
73	
74	\$this->db->set('status', '3');
	\$this->db->where('id_pengaduan', \$id);
	\$this->db->update('pengaduan');
	}
	}
	/* End of file db_pengaduan.php */
	/* Location: ./application/models/db_pengaduan.php */

6.3 Implementasi Algoritme

Pada bagian ini dilakukan implementasi dari perancangan algoritme yang telah dilakukan pada bab sebelumnya. Implementasi dilakukan dengan mengubah algoritme yang dibuat ke dalam kode program dimana kode program dibuat untuk sistem informasi *E-Complaint* Implementasi pada bagian ini hanya beberapa fitur saja.

6.3.1 Verifikasi Pengaduan

Pada proses verifikasi pengaduan terdapat dua pilihan yaitu verifikasi pengaduan memenuhi kriteria dan tidak memenuhi kriteria. Jika pengaduan memenuhi kriteria, sistem akan mengirimkan pesan ke masyarakat “pengaduan memenuhi kriteria”. Jika pengaduan tidak memenuhi kriteria maka pengelola pengaduan harus mengirimkan alasan mengapa pengaduan tidak memenuhi kriteria melalui sistem.

Tabel 6.4 Implementasi verifikasi pengaduan

No. Baris	Source Code
1	public function verifikasi(){
2	
3	if(\$this->input->post('action') == "sukses"){
4	\$id = \$this->input->post('id');
5	
6	\$this->db_pengaduan->verifikasi_sukses(\$id);
7	
8	//kirim pesan
9	\$user = \$this->db_pengaduan->get_pengaduan_detail(\$id)-
10	>row_array();
11	\$pesan = 'Pengaduan Anda dengan ID
12	'.'.\$user['id_pengaduan'].' telah diverifikasi';
13	
14	\$data = array(
15	'pesan' => \$pesan,

No. Baris	Source Code
16	<code>'penerima_id' => \$user['user_id'],</code>
17	<code>'pengirim_id' => \$this->session-</code>
18	<code>>userdata('id_user')</code>
19	<code>);</code>
20	
21	<code>\$this->db_user->send_pesan(\$data);</code>
22	
23	<code>echo \$id;</code>
24	<code>}else{</code>
25	
26	<code>\$id = \$this->input->post('hidden_id');</code>
27	<code>\$alasan = \$this->input->post('alasan');</code>
28	
29	<code>\$this->db_pengaduan->verifikasi_gagal(\$id,\$alasan);</code>
30	
31	<code>// kirim pesan</code>
32	<code>\$user = \$this->db_pengaduan-</code>
33	<code>>get_pengaduan_detail(\$id)->row_array();</code>
34	<code>\$pesan = 'Pengaduan anda dengan ID</code>
35	<code>'.\$user['id_pengaduan'].' tidak memenuhi</code>
36	<code>kriteria.
Alasan: '.\$alasan;</code>
37	
38	<code>\$data = array(</code>
39	<code>'pesan' => \$pesan,</code>
40	<code>'penerima_id' => \$user['user_id'],</code>
41	<code>'pengirim_id' => \$this->session-</code>
42	<code>>userdata('id_user')</code>
43	<code>);</code>
44	
45	<code>\$this->db_user->send_pesan(\$data);</code>
46	
47	<code>echo \$id;</code>
48	<code>}</code>
49	<code>}</code>
50	
51	

6.3.2 Mengirim Pengaduan ke OPD

Pada proses mengirim pengaduan ke OPD pengelola pengaduan harus sudah diverifikasi terlebih dahulu dan memenuhi kriteria. Setelah itu pengelola pengaduan harus memilih pengaduan akan dikirim ke OPD sesuai dengan permasalahan pengaduan. Jika pengaduan sudah dikirim, sistem akan mengirim pesan ke masyarakat pengaduan sudah dikirim ke OPD.

Tabel 6.5 Implementasi mengirim pengaduan ke OPD

No. Baris	Source Code
1	<code>public function forward(){</code>
2	
3	<code>foreach (\$this->input->post('opd') as \$item){</code>
4	
5	<code>\$id = \$this->input->post('hidden_id');</code>
6	
7	<code>\$data = array(</code>
8	<code>'opd_id' => \$item,</code>
9	<code>'pengaduan id' => \$id</code>

No. Baris	Source Code
10);
11	
12	\$this->db_pengaduan->status_dikirim(\$id);
13	\$this->db_pengaduan->insert_forward(\$data);
14	
15	\$user = \$this->db_pengaduan->get_pengaduan_detail(\$id)-
16	>row_array();
17	\$opd = \$this->db_opd->get_opd_byID(\$item)->row_array();
18	
19	\$pesan = 'Pengaduan Anda dengan ID
20	.'. \$user['id_pengaduan'].' telah dikirim ke
21	.'. \$opd['nama_opd'];
22	
23	\$data2 = array(
24	'pesan' => \$pesan,
25	'penerima_id' => \$user['user_id'],
26	'pengirim_id' => \$this->session->userdata('id_user')
27);
28	\$this->db_user->send_pesan(\$data2);
29	}
30	
31	echo \$id;
32	
33	}
34	
35	

6.4 Implementasi Basis Data

Pada bagian ini dilakukan implementasi basis data berdasarkan perancangan PDM yang dibuat pada bab sebelumnya. Tabel-tabel tersebut akan diimplementasikan kedalam bentuk *source code* menggunakan bahasa mysql. Pada sub bab ini akan ditampilkan beberapa hasil implementasi tabel dari perancangan PDM.

6.4.1 Implementasi Tabel Pengaduan

Pada Gambar 5.13 terdapat tabel pengaduan yang memiliki atribut id, status_pengaduan, kategori_id, users_id, judul_pengaduan, isi_pengaduan, tanggal dan relasi dengan tabel users, opd, dan kategori. Tabel tersebut diimplementasikan kedalam bentuk *source code* beserta relasinya. Hasil implementasi tabel pengaduan dapat dilihat pada Tabel 6.6.

Tabel 6.6 Source code tabel pengaduan

No. Baris	Source Code
1	CREATE TABLE `pengaduan` (
2	`id` int(10) UNSIGNED NOT NULL,
3	`status_pengaduan_id` int(10) UNSIGNED NOT NULL,
4	`kategori_id` int(10) UNSIGNED NOT NULL,
5	`users_id` int(10) UNSIGNED NOT NULL,
6	`judul_pengaduan` varchar(100) DEFAULT NULL,

No. Baris	Source Code
7	<code>`isi_pengaduan` varchar(500) DEFAULT NULL,</code>
8	<code>`tanggal` timestamp NULL DEFAULT NULL</code>
9	<code>) ENGINE=InnoDB DEFAULT CHARSET=latin1;</code>
10	
11	<code>ALTER TABLE `pengaduan`</code>
12	<code>ADD PRIMARY KEY (`id`),</code>
13	<code>ADD KEY `pengaduan_FKIndex1` (`kategori_id`),</code>
14	<code>ADD KEY `pengaduan_FKIndex2` (`status_pengaduan_id`),</code>
	<code>ADD KEY `pengaduan_FKIndex3` (`users_id`);</code>

6.4.2 Implementasi Tabel Users

Pada Gambar 5.13 terdapat tabel users yang berelasi dengan tabel jenis_user dan tabel opd. Tabel tersebut diimplementasikan kedalam bentuk source code beserta relasinya. Hasil implementasi tabel pengaduan dapat dilihat pada Tabel 6.7.

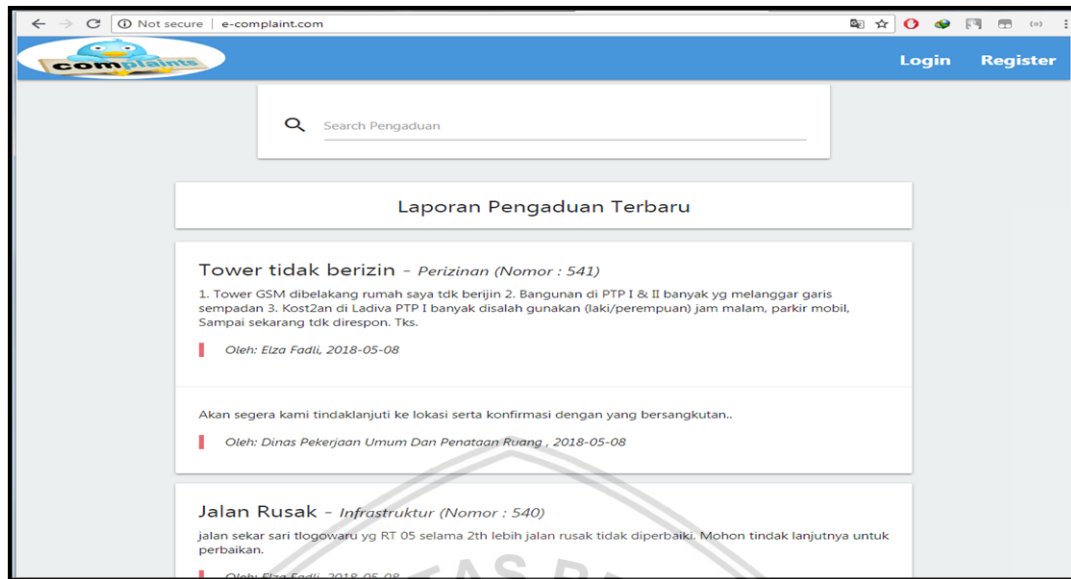
Tabel 6.7 Source code tabel users

No. Baris	Source Code
1	<code>CREATE TABLE `users` (</code>
2	<code>`id` int(10) UNSIGNED NOT NULL,</code>
3	<code>`opd_id` int(10) UNSIGNED NOT NULL,</code>
4	<code>`jenis_user_id` int(10) UNSIGNED NOT NULL,</code>
5	<code>`username` varchar(100) DEFAULT NULL,</code>
6	<code>`passwords` varchar(200) NOT NULL,</code>
7	<code>`nama_lengkap` varchar(100) DEFAULT NULL,</code>
8	<code>`email` varchar(100) DEFAULT NULL,</code>
9	<code>`alamat` varchar(100) DEFAULT NULL,</code>
10	<code>`tanggal_lahir` date DEFAULT NULL,</code>
11	<code>`pekerjaan` varchar(50) DEFAULT NULL</code>
12	<code>) ENGINE=InnoDB DEFAULT CHARSET=latin1;</code>
13	
14	<code>ALTER TABLE `users`</code>
	<code>ADD PRIMARY KEY (`id`),</code>
	<code>ADD KEY `users_FKIndex1` (`jenis_user_id`),</code>
	<code>ADD KEY `users_FKIndex2` (`opd_id`);</code>

6.5 Implementasi Antarmuka

Pada bagian ini dilakukan implementasi antarmuka sistem informasi *E-Complaint* dari perancangan antarmuka pada bab sebelumnya. Pada sub bab ini ditampilkan hasil implementasi dari perancangan tersebut.

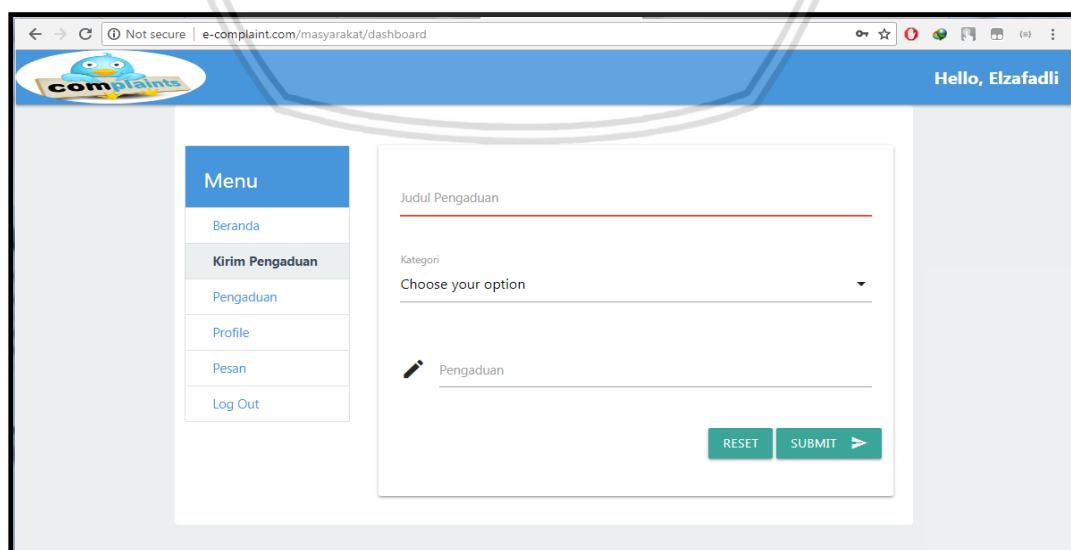
6.5.1 Halaman Awal



Gambar 6.1 Implementasi antarmuka halaman awal

Pada Gambar 6.1 adalah hasil implementasi dari perancangan antarmuka halaman awal *web E-Complaint*. Pada halaman ini masyarakat dapat melihat laporan pengaduan yang sudah ditanggapi atau ditangani oleh OPD di kota XYZ. Masyarakat dapat juga mencari laporan pengaduan menggunakan fitur pencarian yang sudah disediakan. Masyarakat juga dapat mendaftar sebagai user untuk dapat mengirim laporan pengaduan dengan menggunakan menu register. Jika masyarakat sudah mendaftar dapat, masyarakat dapat langsung login dengan menggunakan menu login.

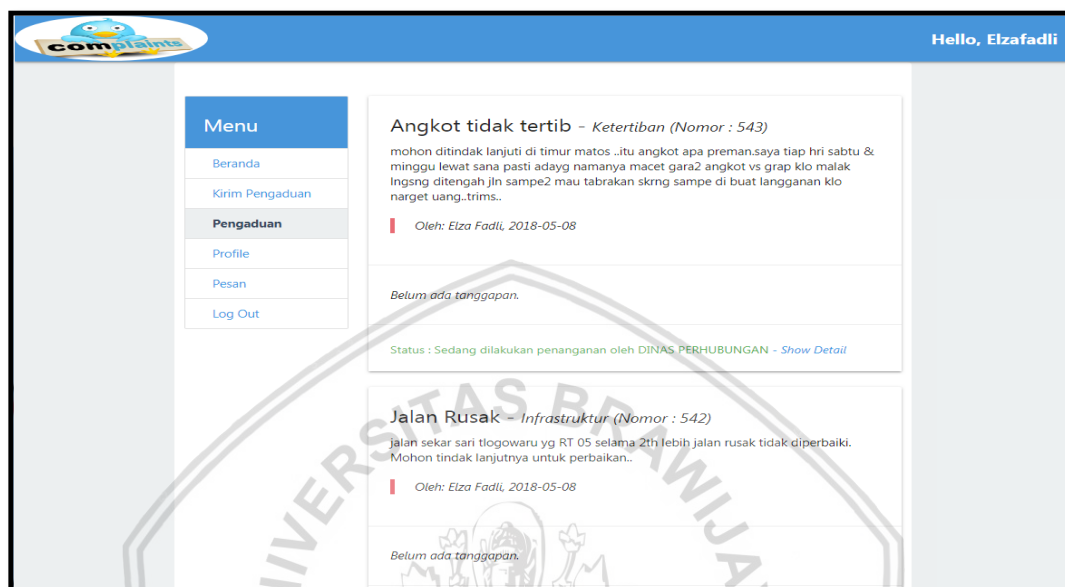
6.5.2 Halaman Mengirim Pengaduan



Gambar 6.2 Implementasi antarmuka halaman mengirim pengaduan

Pada Gambar 6.2 adalah hasil implementasi dari perancangan antarmuka halaman mengirim pengaduan. Pada halaman ini masyarakat dapat mengirim pengaduan mereka melalui *web E-Complaint*. Pada halaman ini masyarakat dapat menggunakan menu yang tersedia seperti melihat status pengaduan.

6.5.3 Halaman Melihat Status Pengaduan

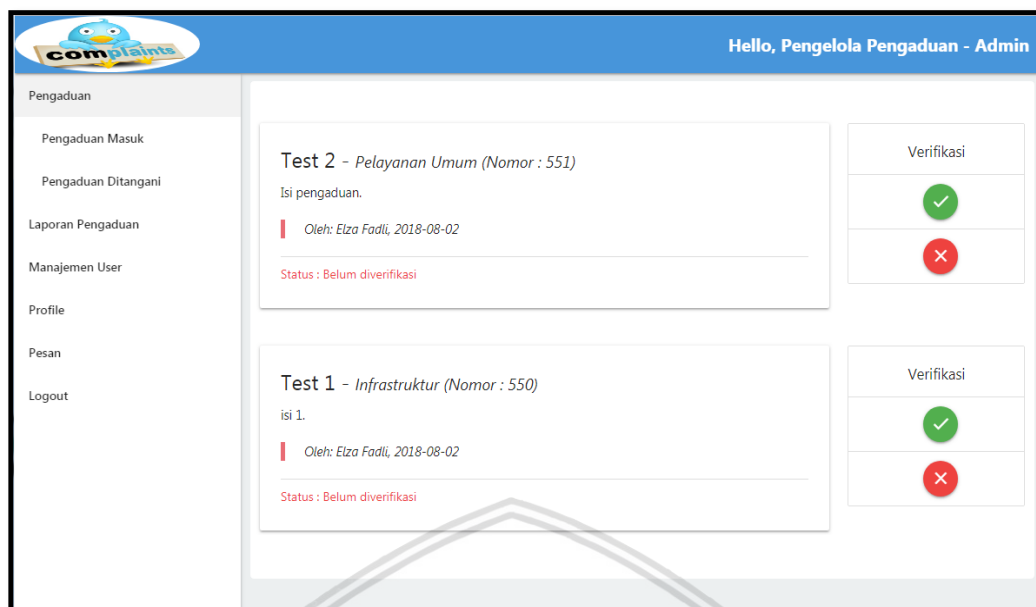


Gambar 6.3 Implementasi antarmuka halaman melihat status pengaduan

Pada Gambar 6.3 adalah hasil implementasi dari perancangan antarmuka halaman melihat status pengaduan. Pada halaman ini masyarakat dapat melihat status pengaduan yang telah mereka kirim.

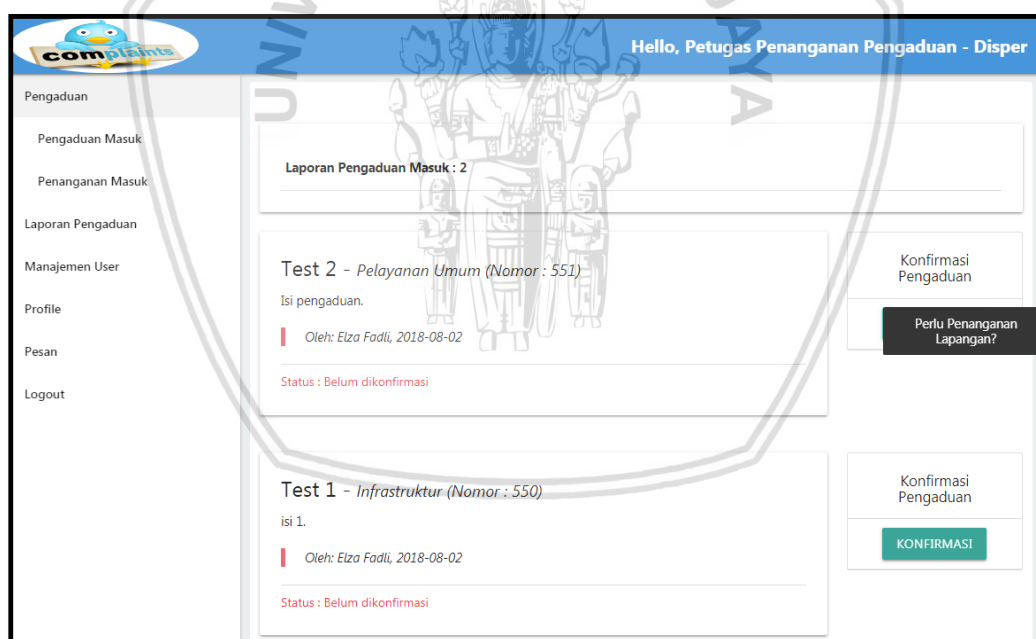
6.5.4 Halaman Verifikasi Pengaduan

Pada Gambar 6.4 adalah hasil implementasi dari perancangan antarmuka halaman verifikasi pengaduan. Pada halaman ini pengelola pengaduan dapat memverifikasi pengaduan apakah memenuhi kriteria atau tidak. Jika pengaduan memenuhi kriteria, sistem akan menampilkan tombol untuk mengirim pengaduan. Jika tidak memenuhi kriteria maka sistem akan menampilkan form untuk mengirim alasan pengaduan tidak memenuhi kriteria.



Gambar 6.4 Implementasi antarmuka halaman verifikasi pengaduan

6.5.5 Halaman Membalas Pengaduan



Gambar 6.5 Implementasi antarmuka halaman membalas pengaduan

Pada Gambar 6.5 adalah hasil implementasi dari perancangan antarmuka halaman membalas pengaduan. Pada halaman ini petugas penanganan pengaduan dapat mengkonfirmasi pengaduan apakah perlu penanganan lapangan atau tidak. Jika perlu penanganan lapangan maka petugas penanganan pengaduan dapat mengirim pengaduan ke petugas di lapangan untuk dilakukan

penanganan. Jika tidak maka, petugas penanganan pengaduan dapat menanggapi pengaduan secara langsung.

6.5.6 Halaman Membuat Laporan

hplaint.com/admin

SHOW GRAPH

Pengaduan Bulan 07 2018

Bulan: Pilih Bulan Tahun: Pilih Tahun

SEARCH

EXPORT TO EXCEL

ID	Kategori	Tanggal Pelaporan	Nama Pelapor	Dikirim Ke	Pengaduan	Tanggapan
549	Fasilitas Umum	2018-07-02 22:48:12	Ahmad Bagas	belum dikirim	Test 3	Sudah dikerjakan saya
548	Fasilitas Umum	2018-07-02 22:47:46	Ahmad Bagas	DINAS PERHUBUNGAN	Test 2	Sudah tanggap kami
547	Pelayanan Umum	2018-07-02 22:47:34	Ahmad Bagas	DINAS PERHUBUNGAN	Test 1	Sudah lakukan penanganan kami

ort?bulan=7&tahun=2018

Gambar 6.6 Halaman membuat laporan

Pada Gambar 6.6 dapat dilihat halaman membuat laporan. Pengguna dapat melakukan sortir laporan pengaduan berdasarkan bulan dan tahun. Setelah itu sistem akan menampilkan hasil sortir pengaduan. Pengguna dapat mengekspor hasil sortir pengaduan ke dalam bentuk excel dengan menekan tombol export to excel. Kemudian akan muncul hasil ekspor dengan bentuk unduhan dalam format excel. Hasil untuk sortir dapat dilihat pada Gambar 6.7.

ID	Kategori	Tanggal Pelaporan	Nama Pelapor	Alamat	Dikirim Ke	Tanggal Pengiriman	Pengaduan	Tanggapan
543	Ketertiban	08/05/2018 07:18	Elza Fadli	Jl. Candi VI B 100 M	DINAS PERHUBUNGAN	08/05/2018 08:09	mohon ditindak lanjuti di timur matos...itu angkot apa preman.saya tiap hri sabtu & minggu lewat sana pasti adayg namanya macet garaz angkot vs grap.klo malak lngsg ditengah jln sampe2 mau tabrakan skrg sampe di buat langganan klo narget uang.trims.	belum ada tanggapan
542	Infrastruktur	08/05/2018 07:18	Elza Fadli	Jl. Candi VI B 100 M	DINAS PERHUBUNGAN	08/05/2018 08:10	jalan sekar sari tlogowaru yg RT 05 selama 2th lebih jalan rusak tidak diperbaiki. Mohon tindak lanjutnya untuk perbaikan.	belum ada tanggapan
541	Perizinan	08/05/2018 00:52	Elza Fadli	Jl. Candi VI B 100 M	DINAS PEKERJAAN UMUM DAN PENATAAN RUANG	08/05/2018 02:07	1. Tower GSM dibelakang rumah saya tdk berijin 2. Bangunan di PTP I & II banyak yg melangar garis sempadan 3. Kost2an di Ladiva PTP I banyak disalah gunakan (laki/perempuan) jam malam, parkir mobil. Sampai sekarang tdk direspon. Tks	Akan segera kami tindaklanjuti ke lokasi serta konfirmasi dengan yang bersangkutan.

Laporan Pengaduan Mei - 2018 (4)

Gambar 6.7 Hasil sortir dalam format excel

BAB 7 PENGUJIAN

Bab ini membahas tahapan pengujian terhadap sistem yang telah dibangun pada penelitian ini. Pengujian sistem ini bertujuan untuk menguji bahwa setiap bagian dari sistem telah sesuai dengan perancangan dan kebutuhan yang telah ditetapkan sebelumnya. Terdapat dua jenis pengujian yang akan dilakukan yaitu pengujian *black-box* untuk menguji kebutuhan fungsional sistem dan pengujian *white-box* untuk menguji tingkat kompleksitas program.

7.1 Pengujian *White-Box*

Pengujian *white-box* dilakukan dengan menggunakan pengujian *basis path testing*. Pengujian dilakukan dengan penggambaran *flow graph* yang berasal dari *pseudocode* atau *flowchart* dari algoritma yang diuji. Kemudian dilakukan penghitungan *cyclomatic complexity* dan jumlah *independent path* yang terdapat pada algoritma yang diuji. Setelah didapatkan jumlah *independent path*, pada pengujian ini dilakukan pemaparan *test case* sesuai dengan jumlah *independent path* yang diperoleh. Kemudian dilanjutkan dengan pengujian integrasi pada beberapa fungsionalitas

Dalam sub bab ini, hanya menggunakan dua kasus pengujian yaitu mengirim pengaduan dan verifikasi pengaduan.

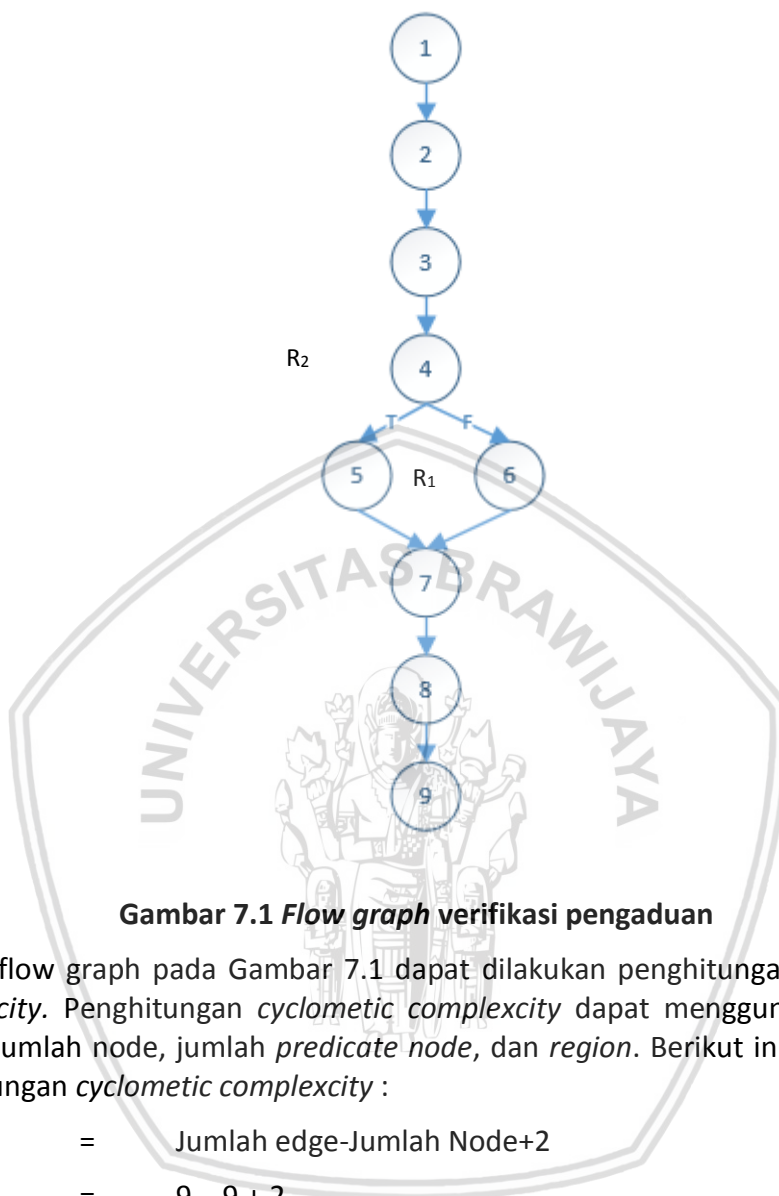
7.1.1 Pengujian Basis Path Verifikasi Pengaduan

Pada Tabel 7.1 adalah *pseudocode* dari algoritma verifikasi pengaduan yang akan digunakan sebagai acuan untuk menentukan *cyclomatic complexity* dan *independent path*.

Tabel 7.1 Perancangan algoritme verifikasi pengaduan

No	Pseudocode
1	Mulai ①
2	Jalankan fungsi verifikasi pengaduan ②
3	Mendapatkan data pengaduan sesuai id yang dipilih ③
4	Jika input memenuhi kriteria ④
5	Mengirim pesan ke masyarakat pengaduan berhasil ⑤
6	deverifikasi
7	Lainnya
8	Mengirim pesan ke masyarakat pengaduan tidak memenuhi
9	kriteria } ⑥
10	Akhir pengkondisian ⑦
11	Menampilkan notifikasi pengaduan berhasil diverifikasi ⑧
12	Selesai ⑨
13	

Flow Graph dihasilkan dari *pseudocode* yang terdapat pada Tabel 7.1 .



Gambar 7.1 Flow graph verifikasi pengaduan

Dari flow graph pada Gambar 7.1 dapat dilakukan penghitungan *cyclometric complexity*. Penghitungan *cyclometric complexity* dapat menggunakan 3 cara melalui jumlah node, jumlah *predicate node*, dan *region*. Berikut ini adalah cara penghitungan *cyclometric complexity* :

$$\begin{aligned}
 V(G) &= \text{Jumlah edge} - \text{Jumlah Node} + 2 \\
 &= 9 - 9 + 2 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Jumlah predicate node} + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Jumlah Region} \\
 &= 2
 \end{aligned}$$

Dari *cyclometric complexity* yang diperoleh didapatkan jalur independent sejumlah 2. Jalur-jalur tersebut antara lain :

- 1) **1-2-3-4-5-7-8-9**

2) 1-2-3-4-6-7-8-9

Berdasarkan 2 jalur tersebut didapatkan *test case* yang dapat dilihat pada Tabel 7.2.

Tabel 7.2 Test case verifikasi pengaduan

No	Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Input memenuhi kriteria	Mengirim pesan ke masyarakat pengaduan berhasil diverifikasi	Mengirim pesan ke masyarakat pengaduan berhasil diverifikasi	valid
2	Input tidak memenuhi kriteria	Mengirim pesan ke masyarakat pengaduan tidak memenuhi kriteria	Mengirim pesan ke masyarakat pengaduan tidak memenuhi kriteria	valid

7.2 Pengujian *Black-Box*

Pengujian *black-box* ini bertujuan untuk memastikan bahwa setiap persyaratan fungsional yang telah didefinisikan sebelumnya mampu dipenuhi oleh sistem yang dibangun. Setiap persyaratan fungsional yang telah didefinisikan pada bagian analisis persyaratan akan diuji pada pengujian ini. Pengujian akan dilakukan dengan mendefinisikan kasus uji (*test case*) yang didasarkan pada skenario untuk setiap *use case* yang telah didefinisikan pada bagian analisis persyaratan. Kemudian setiap hasil pengujian yang dilakukan akan dibandingkan dengan hasil yang telah didefinisikan sebelumnya pada skenario *use case* (hasil yang diharapkan).

Tabel 7.3 Pengujian Validasi

No	Test Name	Test Case	Expected Result	Result	Status
1	Pengujian Mengirim Pengaduan	Pengujian dengan memasukkan input yang benar sesuai format	Sistem menampilkan notifikasi pengaduan berhasil dikirim	Sistem menampilkan notifikasi pengaduan berhasil dikirim	valid
2	Pengujian Mengirim Pengaduan	Pengujian dengan memasukkan input yang salah tidak sesuai format	Sistem menampilkan notifikasi error input salah	Sistem menampilkan notifikasi error input salah	valid

3	Pengujian verifikasi pengaduan	Pengujian dengan memilih opsi memenuhi kriteria	Menampilkan notifikasi pengaduan berhasil diverifikasi dan mengirim pesan ke masyarakat	Menampilkan notifikasi pengaduan berhasil diverifikasi dan mengirim pesan ke masyarakat	valid
4	Pengujian verifikasi pengaduan	Pengujian dengan memilih opsi tidak memenuhi kriteria	Menampilkan notifikasi pengaduan berhasil diverifikasi dan mengirim alasan mengapa pengaduan ditolak	Menampilkan notifikasi pengaduan berhasil diverifikasi dan mengirim alasan mengapa pengaduan ditolak	valid
5	Menyalurkan Pengaduan	Pengujian dengan memilih salah satu OPD	Menampilkan notifikasi pengaduan berhasil diverifikasi dikirim ke OPD	Menampilkan notifikasi pengaduan berhasil diverifikasi dikirim ke OPD	valid
6	Mengidentifikasi Pengaduan	Pengujian dengan memilih opsi perlu penanganan lapangan	Menampilkan notifikasi pengaduan berhasil dikonfirmasi dan menampilkan opsi untuk mengirim laporan pengaduan ke petugas lapangan	Menampilkan notifikasi pengaduan berhasil dikonfirmasi dan menampilkan opsi untuk mengirim laporan pengaduan ke petugas lapangan	valid
8	Mengidentifikasi Pengaduan	Pengujian dengan memilih opsi tidak perlu	Menampilkan notifikasi pengaduan berhasil dikonfirmasi	Menampilkan notifikasi pengaduan berhasil dikonfirmasi	valid

		penanganan lapangan	dan menampilkan opsi untuk menanggapi pengaduan	dan menampilkan opsi untuk menanggapi pengaduan	
9	Mendelegasikan Petugas Lapangan	Pengujian dengan memilih salah satu petugas lapangan	Menampilkan notifikasi pengaduan berhasil dikirim dan mengirim laporan ke petugas lapangan	Menampilkan notifikasi pengaduan berhasil dikirim dan mengirim laporan ke petugas lapangan	valid
10	Mengirim Laporan Penanganan	Pengujian dengan memasukkan input yang benar sesuai format	Menampilkan notifikasi pengaduan berhasil dikirim dan mengirim laporan penanganan ke petugas penanganan pengaduan	Menampilkan notifikasi pengaduan berhasil dikirim dan mengirim laporan penanganan ke petugas penanganan pengaduan	valid
11	Mengkonfirmasi Laporan Penanganan	Pengujian dengan memilih opsi setuju laporan penanganan	Menampilkan notifikasi laporan penanganan berhasil dikonfirmasi dan mengirim pesan ke petugas lapangan	Menampilkan notifikasi laporan penanganan berhasil dikonfirmasi dan mengirim pesan ke petugas lapangan	valid
12	Mengkonfirmasi Laporan Penanganan	Pengujian dengan memilih opsi tidak setuju laporan penanganan	Menampilkan notifikasi laporan penanganan berhasil dikonfirmasi dan menampilkan <i>form</i> untuk	Menampilkan notifikasi laporan penanganan berhasil dikonfirmasi dan menampilkan <i>form</i> untuk	valid

			mengirim alasan mengapa laporan tidak disetujui	mengirim alasan mengapa laporan tidak disetujui	
13	Menanggapi Pengaduan	Pengujian dengan memasukkan input yang benar sesuai format	Menampilkan notifikasi tanggapan pengaduan berhasil dikirim dan mengirim tanggapan pengaduan ke pengelola pengaduan dan masyarakat	Menampilkan notifikasi tanggapan pengaduan berhasil dikirim dan mengirim tanggapan pengaduan ke pengelola pengaduan dan masyarakat	valid
14	Mencari Laporan Pengaduan	Pengujian dengan memasukkan kata kunci secara acak	Menampilkan hasil kata kunci tidak ditemukan	Menampilkan hasil kata kunci tidak ditemukan	valid

7.3 User Acceptance Testing (UAT)

Pengujian UAT adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa sistem yang dikembangkan dapat diterima atau tidaknya oleh pengguna. Hasil dari pengujian UAT digunakan untuk menentukan apakah sistem informasi yang dibangun dapat memenuhi kebutuhan dari pengguna atau tidak. Pengujian UAT dilakukan dengan mempresentasikan hasil implementasi sistem informasi *E-Complaint*. Kemudian mengajukan beberapa pertanyaan kepada admin dari pihak pengelola pengaduan yang bertindak sebagai pengguna.

Pertanyaan-pertanyaan dari pengujian UAT didapatkan dari kebutuhan pengguna yang didapatkan dari tahap analisis kebutuhan. Pertanyaan dari UAT yang diajukan ke pihak pengelola pengaduan dapat dilihat pada Tabel 7.4.

Tabel 7.4 Tabel Pertanyaan UAT

No	Pertanyaan	Tujuan Pertanyaan
1	Sistem informasi dapat dioperasikan dengan mudah dan efektif	Untuk mengetahui sistem informasi yang dibuat mudah digunakan dan efektif

2	Sistem dapat memudahkan masyarakat mengirim dan memantau pengaduan	Untuk mengetahui sistem dapat membantu mempermudah masyarakat mengirim pengaduan dan melacak/memantau pengaduan yang mereka kirim
3	Sistem dapat memudahkan pengelola pengaduan dalam memverifikasi dan mengirim pengaduan ke OPD	Untuk mengetahui sistem dapat memudahkan pihak pengelola pengaduan melakukan verifikasi pengaduan dan penyaluran laporan pengaduan ke OPD lain
4	Sistem dapat memudahkan petugas penanganan pengaduan menanggapi laporan pengaduan dari masyarakat	Untuk mengetahui sistem dapat mempermudah petugas penanganan pengaduan menanggapi pengaduan dari masyarakat
5	Sistem dapat memudahkan pengelola pengaduan dan petugas penanganan pengaduan membuat laporan pengarsipan pengaduan sesuai yang dibutuhkan	Untuk mengetahui sistem dapat membantu pengelola pengaduan dan petugas penanganan pengaduan membuat laporan pengaduan sesuai yang dibutuhkan.
6	Sistem dapat memudahkan petugas penanganan pengaduan mengawasi penanganan pengaduan di lapangan	Untuk mengetahui sistem dapat memudahkan petugas penanganan pengaduan mengawasi penanganan pengaduan di lapangan
7	Sistem dapat memudahkan petugas lapangan mengirim laporan penanganan	Untuk mengetahui sistem dapat memudahkan petugas lapangan mengirim laporan penanganan ke petugas penanganan pengaduan
8	Sistem dapat memudahkan Kepala OPD dan Kepala Unit Pengelola dalam memantau kinerja penanganan pengaduan di kota XYZ	Untuk mengetahui sistem dapat memudahkan Kepala OPD dan Kepala Unit Pengaduan memantau kinerja pengelolaan dan penanganan pengaduan

Hasil dari pengujian UAT yang dilakukan oleh admin dari pihak pengelola pengaduan dapat dilihat pada Tabel 7.5 dan 7.6.

Tabel 7.5 Pengujian UAT admin 1

No	Pertanyaan	Jawaban		Komentar
		Ya	Tidak	
1	Sistem informasi dapat dioperasikan dengan mudah dan efektif	✓		
2	Sistem dapat memudahkan masyarakat mengirim dan memantau pengaduan	✓		Seluruh log bisa dilihat oleh masyarakat
3	Sistem dapat memudahkan pengelola pengaduan dalam memverifikasi dan mengirim pengaduan ke OPD	✓		
4	Sistem dapat memudahkan petugas penanganan pengaduan menanggapi laporan pengaduan dari masyarakat	✓		
5	Sistem dapat memudahkan pengelola pengaduan dan petugas penanganan pengaduan membuat laporan	✓		Laporan bisa diunduh menjadi format excel

	pengarsipan pengaduan sesuai yang dibutuhkan			
6	Sistem dapat memudahkan petugas penanganan pengaduan mengawasi penanganan pengaduan di lapangan	✓		Terdapat status pengerjaan
7	Sistem dapat memudahkan petugas lapangan mengirim laporan penanganan	✓		
8	Sistem dapat memudahkan Kepala OPD dan Kepala Unit Pengelola dalam memantau kinerja penanganan pengaduan di kota XYZ	✓		

Hasil dari pengujian UAT dari admin 1 menunjukan bahwa sistem informasi sudah memenuhi kebutuhan dari pengguna seperti mempermudah masyarakat dalam mengirim dan memantau pengaduan. Mempermudah proses pelaporan dan penanganan pengaduan. Sistem dapat mempermudah proses pembuatan laporan dan memantau kinerja penanganan pengaduan di kota XYZ . Sistem informasi yang dibangun juga mendapat komentar dari admin 1 yaitu sistem dapat menampilkan log dari pengaduan yang dikirim oleh masyarakat, laporan pengaduan dapat diunduh menjadi format excel.

Tabel 7.6 Pengujian UAT admin 2

No	Pertanyaan	Jawaban		Komentar
		Ya	Tidak	
1	Sistem informasi dapat dioperasikan dengan mudah dan efektif	✓		
2	Sistem dapat memudahkan masyarakat mengirim dan memantau pengaduan	✓		
3	Sistem dapat memudahkan pengelola pengaduan dalam memverifikasi dan mengirim pengaduan ke OPD	✓		
4	Sistem dapat memudahkan petugas penanganan pengaduan menanggapi laporan pengaduan dari masyarakat	✓		
5	Sistem dapat memudahkan pengelola pengaduan dan petugas penanganan pengaduan membuat laporan pengarsipan pengaduan sesuai yang dibutuhkan	✓		
6	Sistem dapat memudahkan petugas penanganan pengaduan mengawasi penanganan pengaduan di lapangan	✓		
7	Sistem dapat memudahkan petugas lapangan mengirim laporan penanganan	✓		
8	Sistem dapat memudahkan Kepala OPD dan Kepala Unit Pengelola dalam	✓		

	memantau kinerja penanganan pengaduan di kota XYZ			
--	--	--	--	--

Hasil dari pengujian UAT dari admin 2 menunjukan bahwa sistem informasi sudah memenuhi kebutuhan dari pengguna seperti mempermudah masyarakat dalam mengirim dan memantau pengaduan, mempermudah proses pelaporan dan penanganan pengaduan. Sistem informasi yang dibangun mendapat saran dari admin 2 untuk menambahkan fitur mengirim gambar bagi masyarakat yang ingin melaporkan pengaduan menggunakan tambahan foto.



BAB 8 PENUTUP

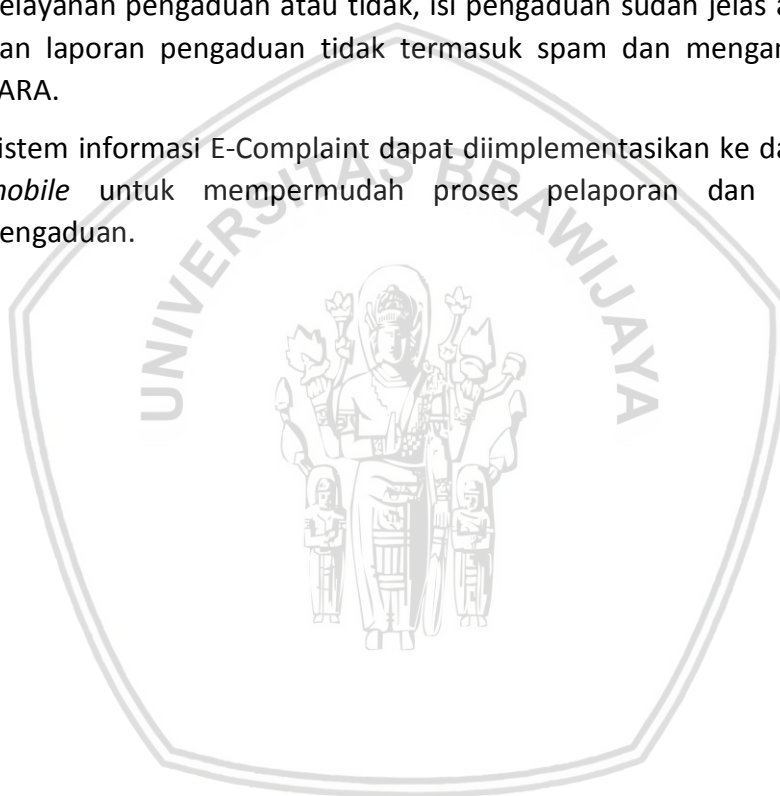
8.1 Kesimpulan

Berdasarkan hasil seluruh penelitian, maka dari penelitian ini dapat diambil kesimpulan sebagai berikut:

1. Dari proses analisis kebutuhan didapatkan identifikasi proses bisnis as-is yang digambarkan menggunakan BPMN, analisis permasalahan, pemodelan proses bisnis to-be menggunakan diagram BPMN, analisis kebutuhan pemangku kepentingan, identifikasi aktor, kebutuhan fungsional dan non fungsional, use case diagram, dan 19 buah spesifikasi *use case* . Sedangkan dari proses perancangan sistem didapatkan 5 buah *activity diagram* untuk menggambarkan alur kerja sistem, *sequence diagram* sejumlah 4 buah untuk menggambarkan interaksi antara sejumlah *object* dengan urutan waktu, *class diagram* untuk menggambarkan kelas-kelas yang akan dibuat di sistem. Dari proses perancangan juga didapatkan perancangan basis data menggunakan ERD dan PDM untuk memudahkan pembuatan tabel-tabel di *database*, perancangan antarmuka, dan perancangan algoritme.
2. Hasil dari implementasi sistem informasi *E-Complaint* didapatkan implementasi *class diagram*, implementasi basis data, implementasi algoritme dan implementasi antarmuka berdasarkan perancangan yang dibuat pada tahap sebelumnya.
3. Sistem informasi *E-Complaint* dapat membuat laporan terkait pengaduan dokumen dengan format Excel. Laporan yang dapat dibuat menggunakan *E-Complaint* meliputi laporan pengaduan setiap bulan, laporan pengaduan OPD tertentu, laporan pengaduan berdasarkan kategori tertentu, laporan pengaduan masuk dan hasil penanganan. Sedangkan untuk OPD menghasilkan laporan pengaduan perbulan khusus OPD, laporan pengaduan khusus OPD berdasarkan kategori tertentu, laporan penanganan khusus OPD.
4. Berdasarkan hasil pengujian UAT dapat disimpulkan sistem informasi *E-Complaint* dapat meningkatkan kualitas pelayanan dan pengelolaan pengaduan di kota XYZ. Dari hasil pengujian UAT sistem dapat meningkatkan kualitas pelayanan pengaduan dengan mempermudah masyarakat mengirim pengaduan dan memantau pengaduan yang mereka kirim. Sistem dapat meningkatkan kualitas pengelolaan pengaduan dengan mempermudah pengelola pengaduan menyalurkan pengaduan ke OPD dan membuat laporan pengarsipan pengaduan, mempermudah OPD dalam menanggapi pengaduan ke masyarakat dan memantau penanganan pengaduan di lapangan.

8.2 Saran

1. Pegujian UAT yang dilakukan di penelitian ini masih belum optimal dikarenakan pengujian UAT hanya dilakukan dari sudut pandang pengelola pengaduan. Untuk penelitian selanjutnya, dapat dilakukan pengujian UAT dari sudut pandang masyarakat dan pihak OPD di kota XYZ untuk mengetahui apakah sistem informasi *E-Complaint* dapat meningkatkan kualitas pelayanan dan pengelolaan pengaduan di kota XYZ.
2. Untuk pengembangan selanjutnya, perlu dilakukan penggunaan metode klasifikasi untuk memverifikasi pengaduan secara otomatis untuk mengetahui apakah pengaduan sudah sesuai dengan ruang lingkup pelayanan pengaduan atau tidak, isi pengaduan sudah jelas atau lengkap dan laporan pengaduan tidak termasuk spam dan mengandung unsur SARA.
3. Sistem informasi E-Complaint dapat diimplementasikan ke dalam aplikasi *mobile* untuk mempermudah proses pelaporan dan penanganan pengaduan.



DAFTAR PUSTAKA

- Anofrizen. 2017. "Sistem Informasi Pengaduan Masyarakat Program Keluarga Harapan Kota Pekanbaru (Studi Kasus : Dinas Sosial Dan Pemakaman Kota Pekanbaru)." *Ilmiah Rekayasa Dan Manajemen Sistem Informasi* 3 (6):97–101.
- Bittner, K. & Spence, L., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Bridgeland, M, and R Zahavi. 2009. "Business Process Modeling Notation," no. January.
- Buttle, Francis. 2008. *Customer Relation Management, Concept and Technologies, Second Edition*. Elsevier Butterworth-Heinemann.
<https://doi.org/10.1007/s13398-014-0173-7.2>.
- IBM. 2012. "Best Practices Physical Database Design for Data Warehouse Environments."
- Masyarakat, Pengaduan. n.d. "Pengertian Dan Bentuk Pengaduan Masyarakat," 11–17.
- Monk, Ellen F., and Bret J. Wagner. 2013. *Concepts in Enterprise Resource Planning. Statewide Agricultural Land Use Baseline 2015*. Vol. 1.
<https://doi.org/10.1017/CBO9781107415324.004>.
- Pressman, Roger S. 2009. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman. Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. <https://doi.org/10.1017/CBO9781107415324.004>.
- Rachmadi P, Adzanil, Fatwa Ramdani, D Sc, S Si, M Sc, Retno Indah Rokhmawati, S Pd, and M Pd. 2017. "Pembangunan Siste ` M Informasi Manajemen Keluhan Pelanggan Berbasis Webgis (Studi Kasus : Outlet Mcdonald ' S Cabang Watu Gong , Malang)" 1 (1):1–5.
- Sidik, B. 2012. "Framework Codeigniter." *Bandung: Informatika*.
https://scholar.google.co.id/scholar?q=Betha+Sidik&btnG=&hl=id&as_sdt=0,5#3.
- Sommerville, Ian. 2010. *Software Engineering. Software Engineering*.
<https://doi.org/10.1111/j.1365-2362.2005.01463.x>.
- Wazlawick, Raul Sidnei. 2014. *Data Persistence Insurance. Neuroinformatics*. Vol. 12. <https://doi.org/10.1007/s12021-014-9239-0>.
- Zanuar, Mohammad Mirza, Mochammad Chandra Saputra, and Fajar Pradana. 2018. "Pengembangan Sistem Informasi KIM (Kelompok Informasi Masyarakat) KOMINFO Jatim Berbasis Web" 2 (1):323–32.